Colour Personal Computer
»CPC464«
Handleiding



CPC464

COLOUR

PERSONAL

COMPUTER

64K

VOORWOORD

Nooit tevoren in de geschiedenis kwamen zoveel nieuwe ontwikkelingen in zo'n korte tijd tot stand, als in de wereld van de computers.

Eigenschappen van computer-hardware en -software hebben een snellere ontwikkeling doorgemaakt dan huidige gebruikers bij machte zijn te volgen.
Daarom ook kan dit handboek voor de CPC 464 niet alle hardware eigenschappen, de volledigheid van zijn operating system en de complete omvang van zijn BASIC beschrijven - dit zou enkele boeken vullen.

Met deze handleiding willen wij u de betekenis van de BASIC instrukties van de Schneider CPC 464 verklaren. Aanvullingen met meer specifieke en gedetailleerde instructieboeken volgen.

Indien u al een ander BASIC dialekt kent, zult u "Schneider BASIC" snel beheersen. Maar ook voor beginners is het goed om te weten dat een duidelijk verstaanbare terminologie de omgang met de CPC 464 vergemakkelijkt.
Daarbij behoren ook realtimefuncties, waarmee veel andere computers niet zijn uitgerust.

Dit handboek is in 3 delen ingedeeld:

Het eerste deel voert de beginner in de wereld van de computertechniek in en verklaart de concepten, begrippen en mogelijkheden van computers.
Degene die tot nu toe nog niet in het bezit was van een personal-computer of nog niet de gelegenheid had daarmee om te gaan, doet er verstandig aan dit deel terdege te bestuderen.

Deel 2 bevat de eigenlijke bedienings-handleiding van de CPC 464 en beschrijft zijn specifieke voordelen.
Wie al enige ervaring heeft met en de begrippen van het werken met computers kent, kan meteen met dit deel, Hoofdstuk 1, beginnen.

Herhaling van grondbeginselen dienen er toe de noodzakelijke handelingen bij de omgang met BASIC beter te leren kennen en een snel inzicht te verschaffen in het werken met geluid en grafics.

Deel 3 tenslotte, is een uitvoerig aanhangsel met een overzicht van de concepten van het werken met computers zowel als een toelichting op de machine-specifieke eigenschappen.

Wij raden u aan ter ondersteuning van deze gebruiksaanwijzing, de te verschijnen BASIC-cursus van Schneider aan te schaffen en door te werken.
Deze biedt u een inleiding in het werken met de CPC 464 en behandeld de nagenoeg grenzeloze mogelijkheden van het systeem als normale computer, zowel educatief als bij het spel.

Ook in de wereld van de computers geldt het gezegde: "proberen gaat boven studeren". U leert het programmeren het eenvoudigst door er mee te werken en u ziet de mogelijkheden, die hij u voor prive en zakelijk gebruik biedt, het beste, wanneer u er mee werkt.

U zult snel tot de ontdekking komen dat gezien zijn prijs geen andere computer u de mogelijkheden biedt van de CPC 464 van Schneider. Daarbij is hij erg gemakkelijk te bedienen. Overigens: Wanneer u ideeen, voorstellen en verbeteringen aan het product (hardware of software) heeft, schrijft u ons dan. Elke reactie is welkom !

Wij wensen u veel plezier en succes.

SCHNEIDER COMPUTER DIVISION.

Alle technische gegevens, informatie en eigenschappen van het in deze gebruiksaanwijzing beschreven product, werden naar beste weten samengesteld en hebben betrekking op de versie, die bij deze druk in de handel is.

Veranderingen en verbeteringen van het product op grond van nieuwe technische ontwikkelingen zijn mogelijk.
Een samenstelling van alle veranderingen en eventuele correcties van dit handboek zijn verkrijgbaar bij Schneider.

Alle onderhoud en service-werkzaamheden moeten door Schneider-dealers uitgevoerd worden. Schneider draagt geen verantwoording voor schade die door ondeskundige bediening resp. service door onbevoegde personen zijn ontstaan. Deze handleiding dient alleen om de gebruiker bij de bediening van zijn produkt te helpen. Schneider neemt geen verantwoording voor schade, die door verkeerde informatie of fouten, resp. foutieve informatie in deze handleiding of door een verkeerde toepassing van het produkt worden gemaakt.

Alle aanwijzingen voor de Z80 in deze handleiding werden met goedkeuring van Zilog Inc. gemaakt.
Eerste uitgave 1984 - orginele uitgave in het Engels.

DE TEKST IN DIT BOEK IS UITGEPRINT MET DE SCHNEIDER NLQ 401 MATRIX PRINTER.

BELANGRIJK !

Wanneer u deze handleiding leest, dient u op de verschillende lettertypes te letten, deze laten zien op welke wijze de informatie moet worden geinterpreteerd : b.v. **[TOETSEN]** op de computer, die geen tekens op het beeldscherm printen, of algemene kommentaren die als verwijzing dienen en niet als instruktie ingevoerd worden.

1 De stekker altijd aan een geaarde wandcontactdoos aansluiten, zoals onder "installeren" beschreven staat.

2 Nooit toetsenbord, monitor of voeding/modulator op andere apparatuur, die niet in deze handleiding wordt genoemd, aansluiten. De apparatuur kan zwaar beschadigd worden en de garantie vervalt.

3 Bloemenvazen, dranken enz. verwijderd houden van monitor, toetsenbord en stroomvoorziening. Door vloeistof wordt in de apparaten grote schade aangericht. In zo´n geval moet gekwalificeerd personeel worden geraadpleegd.

4 Ventilatiegaten op en achter toetsenbord, monitor en voeding/ modulator mogen niet bedekt worden.

5 Wanneer u de stroom uitschakelt, gaat alle informatie in het geheugen van de CPC 464 verloren. In hoofdstuk 2 wordt beschreven hoe een programma bewaard kan worden, nadat u allereerst de grondbeginselcursus heeft doorgenomen.

6 Het is aan te bevelen alleen cassettes, geschikt voor computers, te kiezen. Het is echter mogelijk audio-cassettes van goede kwaliteit te gebruiken (uitgezonderd CrO2, metaal- of C-120- bandjes).
Om een programma sneller te vinden, bevelen wij u C-12 cassettes (6 min. per kant) aan.

7 Cassettes met programma´s van andere computers kunnen in de CPC 464 niet ingeladen, noch worden uitgelezen.

8 De toets **(RECORD)** kan niet ingedrukt worden als de beveiligings- lipjes niet op de cassette aanwezig zijn. Dit verhindert een ongewild overschrijven. Deze toets niet met geweld indrukken, er kan schade aan de apparatuur ontstaan. Om op een aldus beveiligde cassette weer op te nemen, moeten de beveiligingsopeningen worden afgedekt.

9 Zorg er voor dat de cassette de aanloopstrook gepasseerd heeft voordat een programma wordt opgenomen.

10 Stel de apparatuur niet bloot aan de zon, extreme hitte, koude, vocht, stof of aan schokken. Cassettes mogen nooit in de nabijheid van magnetische bronnen komen, bijv. luidsprekers, of grote electrische motoren.

11 Een goede behandeling van de cassettes en het regelmatig schoonmaken van de cassetterecorder zorgen voor een probleemloze opslag en het foutloos terug vinden van een programma.

12 Apparaten niet openmaken, deze bevatten delen die niet door de gebruiker gerepareerd kunnen worden, reparatie mag alleen door technisch personeel geschieden.

INHOUD

Grondbeginselcursus voor beginners.

6 Geluid: introductie.

Geluidsgebied van de CPC464.
Regelingen voor toonhoogte en sterkte.
Wachtlussen voor geluid
Effecten.

7 Printers en Joysticks.

Het gebruik van joysticks.
De Joy-instructie.
Aansluiting van een parallelprinter.

8 Beknopte gids voor CPC464 BASIC.

Een kort overzicht van de BASIC instructies.

9 Verdere informatie over programmeren.

De interne organisatie van de programma's - Firmware.
Onderbrekingen en hun betekenis.
Besturingstekens.
De betrekking tussen machinecode programma's en BASIC instrukties.

10 Onderbrekingsmogelijkheden.

De timer functies.
After, Every en Remain.

AANHANGSELS:

I Voor de beginner: Wat kan de beginner van een computer verwachten en wat niet.
Overzicht van de gebruikte termen.

II Bits en Bytes, binair en hexadecimaal.

III ASCII tekenset.
Tekendefinities en rasters.
Toetsenbordtekens en tokens.

IV Inleiding en overzicht voor gevorderden.

V User-Interface en Expansion-Bus.
Input/output aansluitingen.

VI Beeldscherm-opbouw: plannen en organiseren.

VII Organisatie van het geluid.
Toonhoogte en tijdsduur.

VIII Gereserveerde woorden.
Foutmeldingen.

Schneider CPC464
Introductie cursus
voor Beginners
Grondbeginsel 1:

INSTALLEREN

AANWIJZINGEN VOOR HET UITPAKKEN, AANSLUITEN EN INSCHAKELEN VAN DE CPC464.

De Schneider CPC464 Colour Personal Computer kan naar keuze gekoppeld worden met:

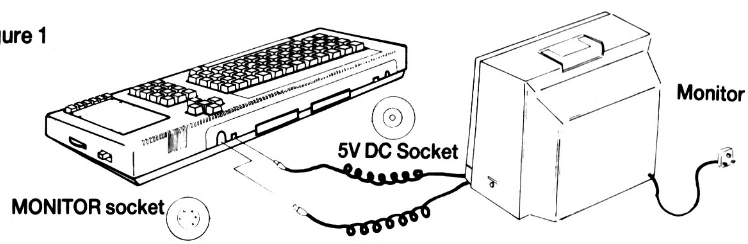
1.1 Schneider GT64 : Groene monitor
1.2 Schneider CTM640 : Kleurenmonitor
1.3 Schneider MP I : Voeding/Modulator en een normaal TV toestel.

Lees de betreffende tekst goed, zodat u uw computer op de juiste wijze aansluit.

1.1 Schneider GT64 Groen monitor

Pak de monitor uit, zet de monitor op een tafel in de buurt van een stopcontact. De tafel moet zo groot zijn dat u de computer gemakkelijk voor de monitor kunt plaatsen. Steek nu de 6 polige DIN stekker die uit de monitor komt, in de met **MONITOR** aangeduide ingang aan de achterkant van de computer (zie tek.1). Verbind de kabel met de kleine stekker (gelijkstroom) uit de monitor met de **5V DC** aangeduide ingang aan de achterkant van de computer.

Figure 1



Zorg er voor dat de AAN/UIT-schakelaar **(POWER)**van de monitor uitgeschakeld is **(OFF)**. Steek de stekker in een 220 VOLT (wisselstroom) stopcontact.

Schakel dan de monitor in en daarna de computer met de schakelaar die zich aan de rechterzijde van de computer **(POWER)** bevindt.

Het rode lampje **(ON)**, dat aan de bovenzijde in het midden zit, brandt nu en op de monitor verschijnt de volgende tekst:

Schneider 64 k Microcomputer (v1)

©1984 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.

BASIC 1.0

Ready
■

CURSOR

Om de ogen niet onnodig te belasten moet U de helderheids-regelaar **(BRIGHTNESS)** zodanig instellen ,dat de tekst niet vervormd is of schaduwranden vertoont.

De contrast-regelaar **(CONTRAST)** zo laag mogelijk instellen.

De regelaar voor verticale synchronisatie **(V-HOLD)** zodanig instellen dat het beeld netjes in het midden van het beeldscherm staat, en niet doorloopt of trilt.

BELANGRIJK !

Wanneer U het systeem niet gebruikt altijd de stekker uit het stopcontact halen.

Omdat intern niets met elkaar doorverbonden mag worden, moet u niet proberen de apparatuur te openen.

1.2 Schneider CTM640 Kleurenmonitor

Pak de monitor uit en plaats hem op een tafel in de nabijheid van een stopcontact, plaats de computer voor de monitor op de tafel.

Verbind, zoals tek.1 laat zien, de kabel met de grote stekker(6-pins DIN) van de monitor, met de met **MONITOR** aangeduide ingang aan de achterzijde van de computer. Steek de kleine stekker van de monitor in de met **5V DC** aangeduide ingang aan de computer.

Vergewis u er van dat de AAN/UIT schakelaar van de monitor op UIT **(OFF)** staat. Steek de stekker van de monitor in een 220 V (wisselstroom) stopcontact.

Schakel de monitor en de computer in met de schakelaar, die zich aan de rechterzijde van de computer bevindt.

BELANGRIJK !

Wanneer u het systeem niet gebruikt, altijd de stekker uit het stopcontact halen.

Omdat intern niets doorverbonden mag worden, moet u niet proberen de apparatuur te openen.

Het rode lampje **(ON)** midden boven op de computer gaat nu branden en op het beeldscherm verschijnt de volgende tekst:

```
Schneider 64K microcomputer(v1)

  ©1984 Amstrad Consumer Electronics plc
            and Locomotive Software Ltd.

  BASIC 1.0

 Ready
 █
```

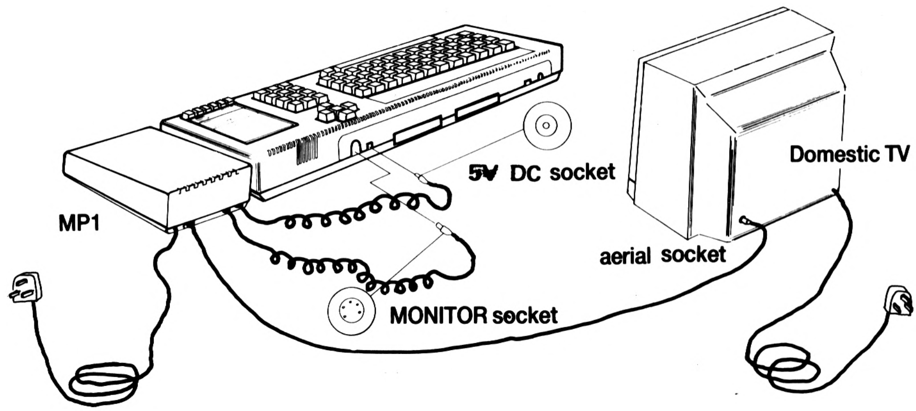
cursor

Om de ogen niet onnodig te belasten, de helderheids-regelaar **(BRIGHTNESS)** zo in stellen dat er geen vervormde of van schaduwranden voorziene tekst ontstaat.

1.3 Schneider MP1 Modulator/voeding en een normaal KTV toestel (UHF)

De MP1 is een randapparaat, dat u misschien gekocht heeft als u een CPC 464 met groen-monitor heeft. Met behulp van de MP1 kunt u uw computer met uw eigen kleuren-televisie verbinden, en zo de kleurmogelijkheden van de CPC 464 benutten.

Pak de MP1 (Modulator/voeding) uit en plaats het rechts van de computer op een tafel in de nabijheid van het T.V. toestel en een stopcontact.



Tek. 2. De MP 1 - Modulator/voeding

Verbind nu de kabel met grote stekker (6 pin DIN), zoals tek. 2 laat zien, van de MP1 met de **MONITOR** ingang aan de achterzijde van de computer. Steek de kleine stekker (gelijkstroom) van de MP1 in de met **5V DC** aangeduide ingang aan de achterzijde van de computer.

Verbind de kabel met de antenneplug van de MP1 met de antenne-ingang van het televisie-toestel.

Vergewis u er van, dat de netschakelaar **(POWER)** aan de rechterzijde van de computer op **OFF** staat en steek dan de stekker van de MP1 in het stopcontact.

Draai de geluidssterkte van het TV toestel helemaal terug, schakel de TV in en schakel dan pas, met de schakelaar aan de rechterzijde **(POWER)**, de computer in.

Het rode lampje **ON** op de computer moet nu gaan branden. U moet nu het TV toestel zo instellen dat een signaal van de computer ontvangen wordt.

Indien u een TV toestel met keuze-schakelaars bezit, druk dan b.v. op een ongebruikte toets. Stel het TV toestel zodanig in dat u ongeveer kanaal 36 ontvangt, waardoor de volgende tekst verschijnt:

```
Schneider 64K Microcomputer (V1)

©1984 Amstrad Consumer Electronics plc
          and Locomotive Software Ltd.

BASIC 1.0

Ready
█
```

Cursor

Regel het TV toestel totdat de tekst duidelijk leesbaar is. De tekst moet goudgeel op een blauwe achtergrond zijn.

Indien uw TV een draaiknop heeft om de kanalen in te stellen, draai deze dan totdat de tekst (zie boven) verschijnt en absoluut rustig blijft.

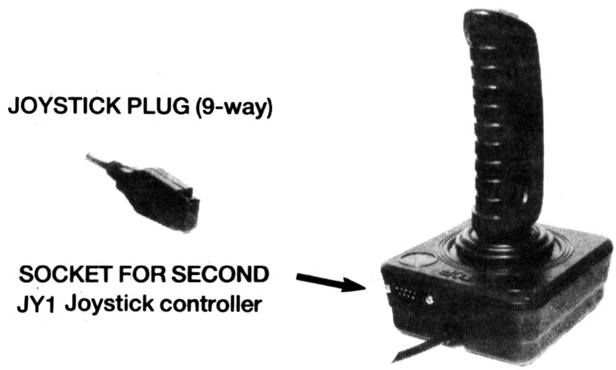
BELANGRIJK !

Trek de stekker altijd uit het stopcontact, wanneer u het systeem niet gebruikt.

Intern mag niets verbonden worden, daarom moet u niet proberen de apparatuur te openen.

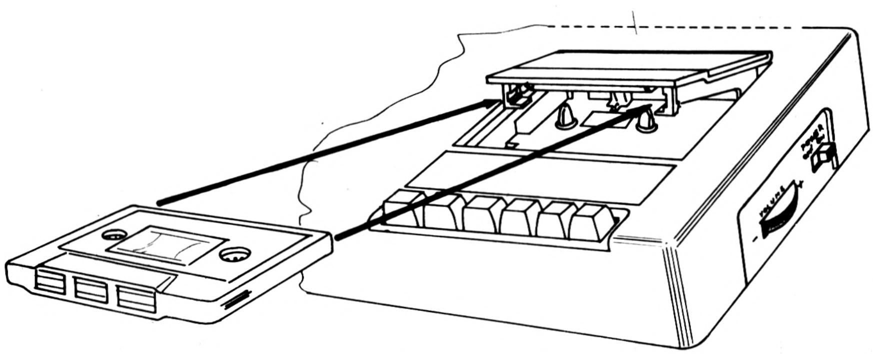
1.4. JOYSTICK

De Schneider joystick JY1 kunt u als een apart onderdeel bij kopen, wanneer u in de CPC464 computerspelletjes laadt die met een joystick gespeeld worden en waarbij ook geschoten wordt. De JY1 wordt aangesloten aan de met **USER PORT (I/O)** aangeduide ingang. De Schneider CPC464 computer kan ook met twee joysticks bediend worden. De tweede joystick kan worden aangesloten aan de eerste joystick.



1.5. De DEMONSTRATIE CASSETTE

In de ene kant van de styropor-verpakking heeft u de demonstratie cassette gevonden. Open het deksel van de datarecorder, door op de toets **[STOP/EJECT]** te drukken, leg dan de cassette in de datarecorder met kant 1 naar boven (zie tek 3).



Tek. 3 Zo legt u de cassette goed in de datarecorder.

Sluit het deksel, druk dan op de **[REW]** toets, totdat de cassette terug gespoeld is. Zo gauw als de band stopt, druk dan op de **[EJECT/STOP]** toets. Zet dan de teller op 000 door op de **COUNTER RESET** knop te drukken.

Druk nu de **[CTRL]** en kleine **[ENTER]** toets, rechtsonder in het cijferblok naast de datarecorder, gelijktijdig in. Op het beeldscherm verschijnt de volgende instructie:

RUN"

Press PLAY then any key:

Druk dan op de **[PLAY]** toets,daarna op een willekeurige letter, nummer, spatiebalk of de **[ENTER]** toets.

De cassette loopt, na korte tijd verschijnt op het beeldscherm:

Loading WELCOME 1 Block 1

Het bandje heeft een laadtijd van circa 5 min.. Op het beeldscherm kunt u volgen hoe het blocknummer verandert in 2,3 enz. totdat de cassette stopt. Het WELCOME programma begint nu te lopen. Het programma begint steeds weer bij het begin, indien u wilt stoppen druk dan tweemaal de **[ESC]** toets, het programma stopt en u kunt op de **[STOP]** toets drukken, de cassette er uit halen, omdraaien en kant 2 inleggen.

Nadat kant 2 ingelegd is denkt u er dan aan dat u op de **[REW]** toets drukt, zodat u er zeker van kunt zijn dat de cassette aan het begin staat.

Druk nu de **[CTRL]** en de kleine **[ENTER]** toets gelijktijdig in. Deze bevindt zich rechts onder in het cijferblok naast de datarecorder. Op het beeldscherm verschijnt het commando

RUN"

Press PLAY then any key:

Druk de **[PLAY]** toets van de datarecorder in en druk op een toets van het toetsenbord. De cassette loopt en na een korte tijd verschijnt de aanwijzing:

Loading WELCOME 2 BLOCK 1

Volgt u de aanwijzing op het beeldscherm op en het programma nodigt u uit om mee te doen, door u commando's te geven, tijdens het RUNnen van het programma.

1.6 Inladen van andere software - cassettes

Het "Welcome" programma kan alleen geladen worden op de manier zoals het in het vorige hoofdstuk beschreven staat.(1.5). Onbeschermde (unprotected) BASIC-programma´s kunnen echter ook anders geladen worden. De nu volgende manier laat dat zien. Spoel het ingelegde bandje terug (**[REW]**-toets op de data-recorder), totdat de spoeltjes zich niet meer bewegen en druk dan de **[STOP/EJECT]**-toets.

Reset de computer om zijn geheugen te wissen. Dat doet u door tegelijk de **[CTRL]** en **[SHIFT]** toetsen in te drukken, ingedrukt te houden en dan op de **[ESC]**-toets te drukken. Daarna laat u alle toetsen weer los. Het beeld wordt gewist en het start-beeld verschijnt net alsof u heeft inschakeld. De instructie **[ENTER]** betekent in de volgende aanwijzingen dat u op een van beide **[ENTER]**-toetsen moet drukken, maar niet het woord **[ENTER]** moet intypen. Het aanhalingsteken krijgt u wanneer u gelijktijdig op de **[SHIFT]** en de 2-toets drukt. (De toets **2** vindt u links in de bovenste toetsenrij)

type in:

Load "" [ENTER]

De computer geeft:

Press PLAY then any key:

Druk nu de **[PLAY]**-toets van de datarecorder, dan op een willekeurige letter-, nummer-,of **[ENTER]**-toets of de spatiebalk.

Het bandje loopt en korte tijd later verschijnt de volgende melding op het beeldscherm:

Loading <programma-naam> **block 1**

De block-nummers gaan met stappen van 1 omhoog tot de cassette geladen is en de melding:

Ready

...op het beeldscherm verschijnt.

U kunt ook de naam van het programma dat u wilt laden, expliciet aan geven, als u intypt:

Load "<titel>" [ENTER]

De computer geeft.

Press PLAY then any key:

Druk nu op de **[PLAY]**-toets van de datarecorder en dan op een willekeurige letter-, cijfer-, **[ENTER]**-toets of spatiebalk.

Het bandje loopt. Wanneer het gewenste programma niet aan het begin van de cassette staat, zoekt de computer de gehele cassette af totdat hij de titel vindt, die u wilt laden. Pas er wel op dat de naam van het programma juist ingetypt wordt.

Indien de computer een andere naam vindt, terwijl hij nog naar het gewenste zoekt, verschijnt de volgende melding:

Found <andere titel> **block 1**

De computer laadt dit programma niet, maar zoekt verder totdat hij het juiste gevonden heeft of totdat u op de **[ESC]**-toets drukt om het zoeken te beeindigen, of tot het bandje afgelopen is.

Als het programma gevonden is verschijnt de melding:

Loading <titel> **Block 1**

De blocknummers worden opgeteld tot het programma helemaal geladen is en de melding:

Ready

...op het beeldscherm verschijnt.

Type dan in:

run [ENTER]

en het zo-even geladen programma loopt. Als een programma al in het geheugen zat, is het gewist en door het nieuwe programma vervangen.

Een programma kan na het laden automatisch gestart worden wanneer u voor het laden intypt:

run "" [ENTER]

De computer antwoordt met:

Press PLAY then any key:

Nadat u de **[PLAY]**-toets en aansluitend een willekeurige toets ingedrukt heeft, zoekt de computer het programma, laadt het en start het zonder verder intikken van instrukties. U kunt dit gebeuren te allen tijde beeindigen door op de **[ESC]**-toets te drukken.

1.7 Het laden van Software-Cassettes. (voorgeprogrammeerd)

U kunt de tot dusver behandelde instrukties gebruiken om een van de vele programma's te laden, die er voor de CPC464 zijn.

Het is raadzaam de laadinstructies op te volgen, die bij elk software product geleverd worden.

1.8 Save

Een programma kan ge**save**d (opgenomen) worden om er later weer over te beschikken.

Legt u een cassette op de goede wijze in de datarecorder en sluit dan het deksel (de beveiligingslipjes tegen wissen mogen niet ontbreken).

Druk de **[REW]**-toets in om de cassette tot het begin terug te spoelen. Wanneer de cassette stopt, druk dan op de **[STOP/EJECT]** toets.Type dan in:

save"<programma-naam> **[ENTER]**

De computer antwoordt met:

Press REC and PLAY then any key:

Druk de **[REC]** en **[PLAY]**toetsen gelijktijdig in, daarna een willekeurige letter-, nummer-, **[ENTER]**-toets of spatiebalk.

De computer toont op het beeldscherm :

saving <programma-naam> **block 1**

Nadat het programma ge**save**d is, stopt de cassette en het woord: **Ready** verschijnt op het beeldscherm.

Druk nu op de **[STOP/EJECT]**-toets van de datarecorder en het programma is ge**save**d.

Denkt u er wel aan dat voorbespeelde software en spelletjes niet op uw eigen cassettes ge**save**d kunnen worden.

Zulke programma's zijn tegen onbevoegd kopieren beschermd.

GRONDBEGINSELEN 2

HET TOETSENBORD

Wij zullen nu de funkties van enkele toetsen van het toetsenbord uitleggen.

[ENTER]

Er zijn twee **[ENTER]** toetsen. Beide geven de informatie die u ingetypt heeft, door aan de computer. Nadat de **[ENTER]** toets is ingedrukt, wordt een nieuwe zin op het beeldscherm begonnen. Elke aanwijzing die wordt ingetypt, moet worden afgesloten met **[ENTER].**

[DEL]

Deze toets wordt gebruikt om een teken of letter, links van de cursor, te wissen.
Typ **a b c d** in en u ziet dat de letter **d** links van de cursor staat. Als u nu de letter **d** niet wilt, drukt u **[DEL]** en ziet dat de **d** verdwijnt. Als u de **[DEL]** toets nu vasthoudt worden alle andere letters ook gewist.

[SHIFT]

De computer heeft twee **[SHIFT]** toetsen. Als u een van deze toetsen ingedrukt houdt en u tikt een letter in, dan wordt een hoofdletter gePRINT of het bovenste teken van de twee op de toets afgebeelde tekens.

Typ **a** in, houdt dan de **[SHIFT]** toets ingedrukt en tik nogmaals een **a**. U ziet dan op het beeldscherm verschijnen:

aA

Geef nu enkele spaties, door de spatiebalk ingedrukt te houden. Gebruik nu de toetsenrij met cijfers, typ het cijfer **2** in, houd dan de **[SHIFT]** toets ingedrukt en typ nogmaals het cijfer **2** in. Op het beeldscherm ziet u nu:

2"

U heeft nu gezien wat er gebeurt wanneer u de **[SHIFT]** toets indrukt en gelijktijdig een tweede toets indrukt. Probeer dit ook met de andere toetsen uit.

[CAPS LOCK]

Deze toets heeft een werking die ongeveer gelijk is aan de **[SHIFT]** toets, deze toets hoeft u echter maar eenmaal in te drukken, want alle ingetypte letters worden hoofdletters, cijfers worden echter niet als symbolen afgebeeld. Druk nu op **[CAPS LOCK]** en typ in:

abcdef 123456

U merkt nu, dat op het beeldscherm de letters in hoofdletters veranderen, de nummers blijven onveranderd. Als u symbolen nodig heeft als de **[CAPS LOCK]** ingedrukt staat, behoeft u alleen de **[SHIFT]** toets te gebruiken als de gewenste symbool toets wordt ingedrukt. Druk de volgende toetsen terwijl u de **[SHIFT]** toets ingedrukt houdt:

ABCDEF123456

Op het beeldscherm verschijnt nu:

ABCDEF!"#$%&

Wanneer u nu weer de kleine letters wilt hebben, houd dan de **[CTRL]** toets ingedrukt en druk op de **[CAPS LOCK]** toets.

ALs u hoofdletters en symbolen wilt typen zonder daarbij de **[SHIFT]** toets steeds te blijven indrukken, houd dan de **[CTRL]** toets ingedrukt en druk op de **[CAPS LOCK]** toets. Type :

abcdef123456

en op het beeldscherm verschijnt:

ABCDEF!"#$%&

Terwijl de **[CTRL]+[CAPS LOCK]** toets in werking zijn kunt u verdere cijfers intypen indien u het toetsenveld rechts van het hoofdtoetsenveld gebruikt.

Wanneer u nu de **[CTRL]** toets ingedrukt houdt en de **[CAPS LOCK]** indrukt wordt de oorspronkelijke modus hersteld. U kunt weer kleine letters krijgen door nogmaals op de **[CAPS LOCK]** toets te drukken.

[CLR]

Deze toets wist een teken onder de cursor.

Typ in: **ABCDEFGH**. De cursor staat rechts van het laatst getypte teken **(H)**. Druk nu de cursor toets links **[←]** viermaal en de cursor staat op de **E**.

Let er wel op hoe de **E** in de cursor nog zichtbaar is. Druk nu op de **[CLR]**-toets en u ziet dat de **E** verdwijnt en de letters **FGH** elk een plaats naar links verschuiven. De letter **F** is nu onder de cursor. Druk nogmaals op de **[CLR]** toets en houd hem vast. U ziet dat de **F** verdwijnt, gevolgd door de **G** en de **H**.

[ESC]

De **[ESC]** toets (ESCAPE) wordt gebruikt om een bewerking af te breken. Druk de **[ESC]** toets in en de computer onderbreekt de bewerking, maar gaat weer verder wanneer een andere toets ingedrukt wordt.
Tweemaal op de **[ESC]** toets drukken heeft tot gevolg dat de computer de lopende bewerking volledig afbreekt. De computer wacht nu op nieuwe instructies van u.

Druk nu tweemaal op de **[ESC]** toets.

BELANGRIJK !

Als u de rechter rand van het scherm bereikt door het intypen van meer dan 40 tekens, verschijnt het volgende teken op de volgende lijn aan de linker rand. U moet niet op de **[ENTER]** toets drukken zoals bij een type-machine gebeurt.
De computer vervult deze functie automatisch voor u en zal bij een onverwachte **[ENTER]** met een foutmelding -bijv. **syntax error** - direkt reageren.

SYNTAX ERROR (syntax fout)

De melding **syntax error** verschijnt op het beeldscherm als de computer niet begrijpt wat u heeft ingetypt. Bijv. typ in:

printt [ENTER]

Op het beeldscherm verschijnt:

syntax error

Deze melding verschijnt omdat de computer **printt** niet begrijpt.
Wanneer u een foutieve programmaregel intypt zoals bv:

10 printt "abc" [ENTER]

verschijnt de melding **syntax error** pas, als de computer deze instructie uitvoert, dus als het programma RUNt. Typ in:

run [ENTER]

Op het beeldscherm verschijnt:

```
syntax error in 10
10 printt "abc"
```

Deze melding zegt u in welke zin de fout is en toont de correctie cursor (EDIT cursor), zodat de fout gecorrigeerd kan worden.
Drukt u op de cursortoets rechts **[→]** tot de cursor op de tweede **t** van **printt** staat. Druk nu op de **[CLR]** toets om de ongewenste **t** te verwijderen en op **[ENTER]** om de gecorrigeerde zin weer in te voeren.
Tik **run [ENTER]** en u ziet dat de computer het commando nu accepteert en **abc** op het beeldscherm PRINT.

Inleiding in Schneider BASIC

U vindt in hoofdstuk 8 een uitvoerige beschrijving van alle BASIC-instructies die in Schneider-BASIC voorkomen. In dit hoofdstuk zullen we enkele gebruikelijke BASIC-instructies bespreken.

CLS

Typ **CLS** in (clear screen: beeldscherm wissen). U kunt hoofdletters of kleine letters gebruiken. Druk dan op **[ENTER]** en u merkt dat het beeldscherm gewist wordt. Het woord **ready** met de cursor ■ verschijnen links boven.

PRINT

Deze instructie wordt gebruikt om tekens, woorden of figuren in het programma op het beeldscherm af te drukken. Typ het volgende in:

print "hallo" [ENTER]

Op het beeldscherm verschijnt dan:

hallo

De aanhalingstekens worden gebruikt om de computer mee te delen, dat er tekst moet worden geprint. **Hallo** verscheen op het beeldscherm, zo gauw **[ENTER]** gedrukt werd. Typ nu **cls** **[ENTER]** om het beeldscherm weer te wissen.

RUN

Het bovenstaande programmavoorbeeld bevat maar een (1) regel. De meeste programma's bestaan uit meerdere programmaregels. Voor elke regel wordt een nummer ingetypt. Deze nummers zeggen de computer in welke volgorde het programma moet RUNnen. Met de **[ENTER]** toets worden de regels een voor een in het geheugen opgeslagen. Typ in:

10 print "hallo" [ENTER]

U merkt dat **hallo** niet op het beeldscherm verscheen, ofschoon **[ENTER]** gedrukt werd. Hiervoor moet u eerst het woord **run** intypen. Typ in:

run [ENTER]

U ziet nu dat **hallo** wel op het beeldscherm verschijnt. In plaats van **print** kunt u het ? gebruiken b.v.:

10 ? "hallo" [ENTER]

LIST

Het is mogelijk een programma te bekijken nadat het ingetypt en in het geheugen opgeslagen is. Typ in:

list [ENTER]

Op het beeldscherm verschijnt:

10 print "hallo"

Dit is het programma dat opgeslagen is in het geheugen.

Let er op dat het woord **PRINT** nu in hoofdletters staat. Dit betekent dat de computer **PRINT** als een BASIC instructie heeft herkend.
Typ nu **CLS [ENTER]**, om het beeldscherm te wissen. Daarbij wordt uw programma niet uit het geheugen gewist ook niet als het op het beeldscherm niet meer te zien is.

GOTO

De **GOTO** instructie vraagt de computer naar een bepaalde regel te springen. Welke regel bepaalt het getal achter **GOTO**. Typ in:

```
10 print "hallo" [ENTER]
20 goto 10 [ENTER]
```

en u ziet hoe **hallo** in een doorlopende kolom op het beeldscherm verschijnt. U kunt het programma onderbreken door op **[ESC]** te drukken. Wanneer u nu op een andere toets drukt, dan loopt het programma weer verder. Het programma kan gestopt worden als u nu tweemaal op **[ESC]**.drukt. Typ in:

cls [ENTER]

om het beeldscherm te wissen.
Om te zien hoe **hallo** van links naar rechts elke regel van het beeldscherm vult, zet u achter het tweede aanhalingsteken van regel 10 van het programma een puntkomma (;). Typ in:

```
10 print "hallo"; [ENTER]
20 goto 10 [ENTER]
run [ENTER]
```

De puntkomma deelt de computer mee dat hij het volgende teken of een groep tekens direkt achter het voorafgaande moet schrijven. U kunt het programma afbreken als u tweemaal **[ESC]** drukt. Voer nu nogmaals regel 10 in maar ditmaal met een komma , in plaats van de puntkomma ;

```
10 print "hallo", [ENTER]
run [ENTER]
```

U ziet dat de komma de computer opdraagt, het volgende teken (of groep) op de 13e positie na het eerste teken van de vorige reeks te schrijven. Deze bewerking is zeer handig wanneer men informatie in verschillende kolommen wil printen. Let er echter op dat het volgende teken 13 spaties verschoven wordt, als het aantal tekens van een groep meer dan 12 bedraagt.

Druk tweemaal **[ESC]** om het programma te onderbreken. Om het computergeheugen helemaal te wissen houdt u achtereenvolgens de **[SHIFT][CTRL] en [ESC]** toetsen ingedrukt. De computer wordt geRESET.

INPUT

Met deze instructie laat u de computer wachten op een invoer, b.v. het antwoord op een vraag. Typ in:

```
10 input "hoe oud bent u"; leeftijd [ENTER]
20 print "u ziet er jonger uit dan "; leeftijd [ENTER]
run [ENTER]
```

Op het beeldscherm ziet u:

Hoe oud bent u?

Voer uw leeftijd in en druk op **[ENTER]**. Wanneer u 18 bent verschijnt:

U ziet er jonger uit dan 18.

Dit voorbeeld laat zien hoe men **input** en een numerieke variabele gebruikt. Het woord **leeftijd** werd aan het einde van regel 10 in het geheugen gebracht, zodat de computer het woord **leeftijd** met het ingevoerde getal in verband kon brengen. Alhoewel wij de variabele **leeftijd** voor ons voorbeeld hebben gebruikt konden we ook net zo goed de letter **b** gebruiken.

RESET de computer, om het geheugen te wissen, met de **[CTRL],[SHIFT] en [ESC]** toetsen. Als u een invoer (**input**) van letters en eventueel cijfers wenst, moet u achter de variabelenaam het dollarteken **$** toevoegen. Typ het volgende programma in: (Let er op, in regel 20 na de **o** in **hallo** en voor de **m** in **mijn** een spatie toevoegen.)

```
10 input "Hoe heet u "; naam$ [ENTER]
20 print "hallo "; naam$" mijn naam is Arnold" [ENTER]
run [ENTER]
```

Op het beeldscherm ziet u:

Hoe heet u ?

Voer uw naam in (bv Fred) en druk op **[ENTER]**. Op het beeldscherm verschijnt:

hallo Fred mijn naam is Arnold
(Misschien interesseert het u dat **Arnold** de codenaam voor de Schneider CPC464 was tijdens de ontwikkeling.)

Wij hebben **naam$** voor de stringvariabele gebruikt; We hadden ook **a$** kunnen gebruiken. RESET de computer met **[CTRL],[SHIFT] en [ESC]** toetsen en typ in:

```
 5 cls [ENTER]
 10 input "Wat is uw naam"; a$ [ENTER]
 20 input "Wat is uw leeftijd"; b [ENTER]
 30 print "Ik moet zeggen ";a$ ", u ziet er niet als een";b "jarige
uit" [ENTER]
 run [ENTER]
```

Wij hebben in dit programma twee variabelen gebruikt, **a$** voor de naam en **b** voor de leeftijd. Op het beeldscherm ziet u:

Wat is uw naam?

Voer uw naam in en druk op **[ENTER]**. Dan wordt u gevraagd:

Wat is uw leeftijd?

Voer uw leeftijd in (bv 18) en druk op **[ENTER]**. Als u Fred heet en 18 bent ziet u nu op het beeldscherm:,

Ik moet zeggen Fred, u ziet er niet als een 18 jarige uit.

EDITTEN VAN EEN PROGRAMMA

Fout ingevoerde programmaregels, die syntax- of andere fouten hebben, kunnen gecorrigeerd worden in plaats van ze opnieuw in te typen. Om te laten zien hoe dit gaat: typ het vorige programma in met fouten.

```
 5 clss [ENTER]
 10 input "Wat is u naam"; a$ [ENTER]
 20 input "Wat is uw leeftijd"; b [ENTER]
 30 input "Ik moet zeggen"; a$", u ziet er niet als een"; b"jarige
uit" [ENTER]
```

We hebben 3 fouten in het programma gemaakt.
In regel 5 hebben wij **clss** i.p.v. **cls** getypt.
In regel 10 hebben wij **u** i.p.v. **uw** getypt.
In regel 30 hebben wij de spatie tussen **zeggen** en " vergeten.

We hebben 3 mogelijkheden een programma te verbeteren. De eerste is: de regel helemaal opnieuw in typen. Als de regel ingetypt wordt, vervangt deze de regel in het geheugen met hetzelfde nummer. De tweede mogelijkheid is de **EDIT** methode en ten derde hebben we nog de zogenaamde **COPY Cursor** methode.

EDIT methode

Om de fouten in regel 5 te corrigeren, tikken we in:

edit 5 [ENTER]

Regel 5 wordt afgedrukt met de cursor op de 5. Druk op de rechter cursortoets [→] totdat de cursor op de **s** staat, die teveel is in **clss**, druk nu op de **[CLR]**toets en de **s** verdwijnt.
Druk vervolgens op de **[ENTER]** toets en de gecorrigeerde regel 5 wordt in het geheugen opgeslagen.

Typ in:

list [ENTER]

om te zien of de correctie goed is uitgevoerd.

COPY Cursor methode

Om de fouten in regel 10 en 30 te corrigeren drukt u gelijktijdig op de **[SHIFT]** en de cursor-omhoog-toets [↑] tot de COPY-Cursor aan het begin van regel 10 staat. Druk dan op de **[COPY]**toets tot de COPY-Cursor op de spatie tussen **u** en **naam** staat. U merkt dat regel 10 nogmaals op de onderste regel wordt geschreven en de hoofd-cursor op dezelfde plaats stopt als de COPY-Cursor. Typ dan een **w** in. Deze wordt op de onderste regel toegevoegd.
De hoofdcursor heeft zich bewogen, maar de COPY-Cursor is blijven staan. Druk nu de **[COPY]**toets tot de regel 10 helemaal is overgenomen. Druk **[ENTER]** en de nieuwe regel 10 wordt in het geheugen opgenomen. De COPY-Cursor verdwijnt en de hoofd cursor staat nu onder de nieuwe regel 10.

De andere fout kan op een gelijkwaardige wijze gecorrigeerd worden, druk gelijktijdig op de **[SHIFT]** en cursor-omhoog-toets [↑] tot de COPY-Cursor aan het begin van regel 30 staat.
Druk op de **[COPY]**toets tot de COPY-Cursor op het aanhalingsteken naast **zeggen** staat. Druk eenmaal op de spatiebalk, een spatie wordt in de onderste regel toegevoegd. Druk nogmaals op de **[COPY]** toets tot regel 30 compleet is overgenomen. Druk vervolgens op de **[ENTER]** toets. U kunt het hele programma controleren indien u **list [ENTER]** intypt.

RESET de computer met [CTRL], [SHIFT] **en [ESC].**

IF THEN

Wij breiden nu ons programma uit door twee nieuwe instructies: **if** en **then** toe te voegen.

Typ het volgende in: (let op dat we twee symbolen hebben toegevoegd < betekent kleiner dan en zit naast de **m** toets, > betekent groter dan en zit naast de < toets.

```
 5 cls [ENTER]
 10 input "Wat is uw naam"; a$  [ENTER]
 20 input "Wat is uw leeftijd"; leeftijd [ENTER]
 30 if leeftijd < 13 then 60 [ENTER]
 40 if leeftijd < 20 then 70 [ENTER]
 50 if leeftijd > 19 then 80 [ENTER]
 60 print "Wel, "; a$ " U bent met" leeftijd "jaar nog geen teenager":
    end [ENTER]
 70 print "Dus " a$ " is een teenager met" leeftijd "jaar":end [ENTER]
 80 print "Nou, " a$" met" leeftijd "bent u geen teenager meer"[ENTER]
```

Controleer het programma met:

List [ENTER]

typ vervolgens in :

run [ENTER]

Beantwoord nu alle vragen van de computer en u ziet wat er gebeurt.

U ziet nu welke funkties **if** en **then** in het programma hebben. Wij hebben ook het woord **END** aan het einde van regel 60 en 70 toegevoegd. De instructie **END** betekent letterlijk: eindig het programma. Zonder **END** in regel 60 zou het programma verder lopen en de regels 70 en 80 afwerken. Evenzo zou zonder **END** aan het einde van regel 70 het programma verder lopen en regel 80 uitvoeren. De dubbele punt (:) geeft u de mogelijkheid twee of meer statements in een regel te schrijven. De dubbele punt voor de instructie **END** scheidt het van de voorafgaande statement. We hebben ook een regel 5 ingevoegd, om het beeldscherm aan het begin van het programma te wissen. We zullen dit van nu af aan in de volgende programma's altijd doen, zodat ons werk er beter uitziet.

Andere sleutelwoorden in verbinding met **IF** en **THEN** zijn o.a. **ELSE**, **OR** en **GOTO**.

RESET de computer met **[CTRL][SHIFT]en[ESC]**

FOR TO NEXT

Wij laten nu het gebruik van de **FOR**, **TO** en **NEXT** instructies zien.
In een voorbeeld laten wij zien hoe de computer de tafel van twaalf (1*12, 2*12 enz.) kan uitvoeren.
In het voorbeeld is * het teken van een vermenigvuldiging.

```
5 cls [ENTER]
10 for a=1 to 20 [ENTER]
20 print a"* 12 ="a*12 [ENTER]
30 next a [ENTER]
run [ENTER]
```

U merkt dat de kolommen niet netjes zijn gerangschikt, typ daarom in:

```
5 cls [ENTER]
10 for a= 1 to 9 [ENTER]
20 print a" * 12 ="a*12 [ENTER]
30 next a [ENTER]
40 for a= 10 to 20 [ENTER]
50 print a"* 12 ="a*12 [ENTER]
60 next a [ENTER]
run [ENTER]
```

Probeer dit programma eens met andere getallen, b.v. 17. Daarvoor moet u "12" in regel 20 en 50 door "17" vervangen. Bent u klaar, RESET dan de computer met **[CTRL][SHIFT] en [ESC]**. U kunt de stapgrootte in FOR TO NEXT met STEP specificeren. Verdere informatie hierover vindt u in hoofdstuk 8.

EENVOUDIGE REKENKUNDE

Uw CPC464 kan ook als tafel-rekenmachine worden gebruikt.

De volgende voorbeelden laten dit zien. Wij gebruiken in dit deel ? in plaats van **PRINT**. Het antwoord wordt gegeven als de **[ENTER]** toets wordt ingedrukt.

OPTELLEN

(Druk voor de + de **[SHIFT]** en de ; toets.)

Typ in:

```
?3+3 [ENTER]
6
```

(het = teken wordt niet ingetypt.)

Typ in:

```
?8+4 [ENTER]
12
```

AFTREKKEN

(Gebruik het = teken zonder **[SHIFT]** voor -)

Typ in:

```
?4-3 [ENTER]
1
```

Typ in:

```
?8-4 [ENTER]
4
```

VERMENIGVULDIGEN

(Gebruik de **[SHIFT]** en : toets voor vermenigvuldigen. * betekent x)

Typ in:

```
?3*3 [ENTER]
9
```

Typ in:

```
?8*4 [ENTER]
32
```

DELEN

(Gebruik de ? toets zonder **[SHIFT]** om te delen. / betekent ÷.)

Typ in:

```
?8/4 [ENTER]
2
```

WORTELTREKKEN

U berekent de wortel van een getal door SQR(getal) in te typen.
Typ in:

```
?sqr(16) [ENTER]
 4
```

Typ in:

```
?sqr(100) [ENTER]
 10
```

MACHTSVERHEFFEN

(Gebruik de £ toets zonder [SHIFT] voor machtsverheffen.)

Het tweede getal bepaalt, hoe vaak het eerste met zichzelf moet worden vermenigvuldigd. bijv. $3^2 = 3x3$, $3^3 = 3x3x3$
Typ in:

```
?3↑3 [ENTER] (dit betekent 3³ )
 27
```

tik in:

```
?8↑4 [ENTER] (dit betekent 8⁴ )
 4096
```

DERDEMACHTSWORTELS

De derdemachtswortel wordt met dezelfde methode berekend als die bij machtsverheffen.
U vindt de derdemachtswortel van 27($\sqrt[3]{27}$) als u intypt:

```
?27↑(1/3) [ENTER]
 3
```

De derdemachtswortel van 125:

```
?125↑(1/3) [ENTER]
 5
```

SAMENGESTELDE BEREKENINGEN

Samengestelde berekeningen kunnen met de computer worden uitgevoerd, volgens de geldende regels.

De hoogste prioriteit heeft de vermenigvuldiging, dan het delen, dan de optelling en als laatste het aftrekken. Deze prioriteiten gelden alleen als deze vier berekeningsmethoden gebruikt worden. De prioriteiten worden later verder besproken, dan inclusief de machten

Neem de volgende berekening:
3+7-2*7/4

Misschien denkt u dat het zo berekend wordt:
3+7-2*7/4
= 8+7/4
= 56/4
= 14

In werkelijkheid wordt zo gerekend:
3+7-2*7/4
=3+7-14/4
=3+7-3,5
=10 -3,5
=6,5

U kunt dit uitproberen door het volgende voorbeeld in te typen:
?3+7-2*7/4 [ENTER]
6.5

De volgorde van de berekening kan veranderd worden door haakjes () te gebruiken. De bewerking tussen haakjes heeft voorrang op de bewerking buiten de haakjes. Overtuigt u zich ervan door de volgende berekening met haakjes:
?(3+7-2)*7/4 [ENTER]
14

MACHTSVERHEFFEN

Als u zeer grote of zeer kleine getallen nodig heeft, is een andere schrijfwijze voordeliger. De letter E duidt een macht van 10 aan. U kunt zowel de hoofdletter als de kleine letter E gebruiken. Het getal 300 kan ook als **3x10↑2** geschreven worden. In de wetenschappelijke schrijfwijze schrijft men 3E2. Hetzelfde geldt voor **0,03: 3x10↑-2**. Met de wetenschappelijke notatie krijgen we 3E-2. Probeer het volgende voorbeeld uit:

1. 30x10

U kunt intypen:
?30*10 [ENTER]
300

Of:
?3e1*1e1 [ENTER]
300

2. 3000*1000

Typ in:
?3e3*1e3 [ENTER]
3000000

3. 3000x0,001

Typ in:
?3e3*1e-3 [ENTER]
3

GRONDBEGINSELEN 3
Graphics, mode en geluid

Op het beeldscherm van de CPC464 Colour Personal Computer zijn 3 verschillende modi mogelijk: mode 0, mode 1 en mode 2.

Na het inschakelen van de computer is automatisch mode 1 ingesteld. Om de verschillende modi te begrijpen schakelen we de computer en drukken we op de toets met 1. Houd de toets ingedrukt tot 2 regels vol met enen staan. Wanneer u deze enen gaat tellen zult u zien dat er op elke regel 40 staan. D.w.z. dat in mode 1 veertig posities beschikbaar zijn. Druk nu op **[ENTER]** - U krijgt wel een **Syntax error**, maar die stoort ons niet. We komen alleen sneller op **Ready** terug, zodat de volgende instructie gegeven kan worden. Typ in:

mode 0 [ENTER]

U ziet dat de tekens op het beeldscherm nu groter zijn. Druk nu weer op de 1 toets tot 2 regels volgeschreven zijn. Er staan nu op elke regel 20 enen, d.w.z. dat we in Mode 0 maar 20 posities beschikbaar hebben. Druk nu nogmaals op **[ENTER]**. Typ nu in:

mode 2 [ENTER]

Dit is de kleinste modus en wanneer u nu een regel met enen intypt dan ziet u 80 cijfers. In mode 2 hebben we 80 posities. Nogmaals:

Mode 0: 20 posities
Mode 1: 40 posities
Mode 2: 80 posities

Druk tenslotte nog eens op **[ENTER]**.

KLEUREN

We kunnen kiezen uit 27 kleuren, deze verschijnen op een groen beeldscherm (GT64) als verschillende tinten groen. Wanneer u de monitor GT64 bezit, kunt u een Schneider MP1 modulator erbij kopen om de kleurmogelijkheden van uw computer op uw TV toestel te kunnen gebruiken.

In mode 0 kunt u 16 van de 27 beschikbare kleuren tegelijk gebruiken.
In mode 1 kunt u 4 van de 27 kleuren tegelijk gebruiken.
In mode 2 kunt u 2 van de 27 kleuren tegelijk gebruiken.

U kunt de kleuren van de beeldrand **(BORDER)**, van het werkvlak (**PAPER** - de vlakte waarop de tekens staan) of van de **PEN** (het teken zelf] onafhankelijk van elkaar veranderen.

De 27 mogelijke kleuren zijn met de daarbij behorende **INK** nummers in tabel 1 afgebeeld.

KLEURENTABEL

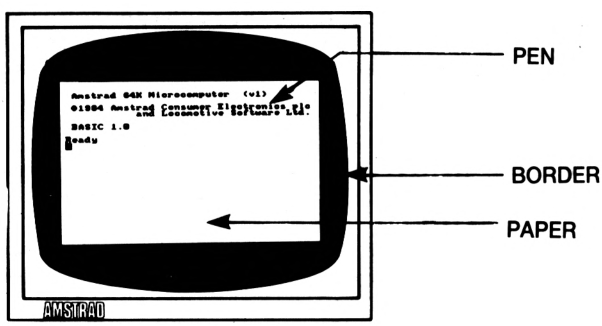
INK	KLEUR	INK	KLEUR
0	zwart	14	pastelblauw
1	blauw	15	oranje
2	licht blauw	16	rose
3	rood	17	pastelmagenta
4	magenta	18	lichtgroen
5	lichtpaars	19	zeegroen
6	lichtrood	20	lichtcyaan
7	purper	21	geelgroen
8	lichtmagenta	22	pastelgroen
9	groen	23	pastelcyaan
10	cyaan	24	lichtgeel
11	hemelsblauw	25	pastelgeel
12	geel	26	wit
13	grijs		

Zoals reeds gezegd, start de computer na het inschakelen in mode 1. Om mode 1 terug te krijgen typt u in:

mode 1 [ENTER]

De kleuren van **BORDER, PAPER en PEN** bij het inschakelen zijn:

BORDER(rand) : nr.1 (blauw)
PAPER(beeldscherm): nr.1 (blauw)
PEN(teken) : nr.24 (lichtgeel)



De rand (**BORDER**) is de zone rond het werkvlak (**PAPER**). Na het inschakelen zijn zowel **BORDER** als **PAPER** blauw. **PAPER** (papier) is het werkvlak binnen de **BORDER**(rand), waarop de tekens kunnen staan. **PEN** is de kleur van de tekens.

Dit wordt misschien duidelijk als we de PEN (cursor) en PAPER (werkvlak) met een tekenpen en een blad papier vergelijken. Net zoals de kleur van de inkt (INK) in een tekenpen (PEN) kunnen wijzigen zo kunnen we ook de kleur van de tekens op het beeldscherm wijzigen. Het papier kan verschillende kleuren hebben. Zo is het ook met de kleur van het werkvlak (PAPER) van het beeldscherm.
De kleur van de rand(BORDER) kunt u veranderen als u intypt:

border 0 [ENTER]

De kleur van de rand(BORDER) verandert van blauw in zwart. In tabel 1 ziet u dat 0 voor zwart staat. U kunt de randkleur in elk van deze kleuren veranderen door BORDER en dan het kleurnummer in te typen.
Typ nu in:

cls [ENTER]

om het beeldscherm te wissen.

Wanneer u de kleur van het werkvlak (PAPER) wilt veranderen, typt u in:

paper 2 [ENTER]

De achtergrondkleur achter het woord **Ready** is nu licht cyaan.
Typ in:

cls [ENTER]

om het werkvlak met de nieuwe kleur te vullen. U kunt ook de kleur van de PEN veranderen:

pen 3 [ENTER]

De kleur van de PEN is nu veranderd en het woord **Ready** is lichtrood. Typ nu in:

cls [ENTER]

U zult het bovenstaande beter begrijpen als u tabel 2 ook bekijkt. Na het inschakelen heeft PAPER het nummer 0. In tabel 2 vindt u in de eerste kolom onder PEN het nummer 0. In dezelfde regel onder mode 1 vindt u kleur nummer 1.

Wanneer u kleurtabel 1 bekijkt, vindt u onder nummer 1 de kleur blauw, deze kleur heeft het PAPER als u de computer inschakelt.

Wij hebben net PAPER op nummer 2 gezet. In de linker kolom in tabel 2 vindt u PAPER nummer 2: in dezelfde regel vindt u onder mode 1 het kleurnummer 20. Met tabel 1 kunt u aflezen dat 20 voor lichtcyaan staat.

	Ink Colour		
Paper/Pen No.	Mode 0	Mode 1	Mode 2
0	1	1	1
1	24	24	24
2	20	20	1
3	6	6	24
4	26	1	1
5	0	24	24
6	2	20	1
7	8	6	24
8	10	1	1
9	12	24	24
10	14	20	1
11	16	6	24
12	18	1	1
13	22	24	24
14	Flashing 1,24	20	1
15	Flashing 16,11	6	24

Table two: PAPER/ PEN/MODE/INK reference



Nadat de computer ingeschakeld is, heeft de PEN nr.1. Als u tabel 2 bekijkt ziet u dat in mode 1 voor PEN kleurnummer 24 staat. In tabel 1 staat voor 24 de kleur lichtgeel. Dit is ook de kleur, die de tekens op het beeldscherm hebben na het inschakelen.

We hebben daarnet PEN op 3 gezet. In tabel 2 ziet u dat PEN 3 overeenkomt met kleur 6 in mode 1. In tabel 1 ziet u dat de kleur 6, lichtrood is.

Wij gebruikten daarnet PAPER 2 en PEN 3. Deze kleuren kunnen we verder veranderen. Dit doen we d.m.v. de **INK** instructie. De INK instructie gebruikt twee waarden; het eerste zegt welke PEN of PAPER nr. veranderd moet worden. Het tweede geeft de nieuwe kleur voor PEN of PAPER aan. Bekijk in tabel 1 de kleurnummers. Als voorbeeld veranderen we de kleur van PAPER 2 in zwart en de kleur van PEN 3 in wit.

Uit tabel 1 leren we dat zwart nr.0 en wit nr.26 is.

Typ in:

ink 2,0 [ENTER]

(als uitgelegd is 2 het huidige **PAPER** nummer, en 0 is zwart.)

Typ in:

ink 3,26 [ENTER]

(3 is het huidige **PEN** nummer en 26 is wit.)

RESET nu de computer door achter elkaar de **[CTRL],[SHIFT] en [ESC]** toetsen in te drukken en vast te houden.

Zoals hierboven vermeld, heeft na het inschakelen van de computer of na het RESETten (met **[CTRL], [SHIFT]** en **[ESC]** toetsen) **PAPER** de waarde 0 **en** PEN **nr.1**. De kleur van PAPER is 1 (blauw) en de kleur van PEN is 24 (lichtgeel). Deze moeten als INK 0,1 voor PAPER en INK 1,24 voor de PEN ingegeven worden. Om deze direkt in een wit teken (PEN) op een zwarte achtergrond(PAPER) te veranderen,dient u in te typen:

ink 0,0 [ENTER]

en dan:

ink 1,26 [ENTER]

KNIPPERENDE KLEUREN

We kunnen de tekens in twee verschillende kleuren, laten knipperen. Deze functie krijgen we door een tweede kleurnummer in de INK instructie voor PEN aan te geven.

U ziet de tekens op het beeldscherm afwisselend wit en lichtrood als u de volgende instructie intypt:

ink 1,26,6 [ENTER]

In dit geval is 1 het PEN nummer, waarbij 26 wit en 6 de alternatieve kleur lichtrood zijn.

We kunnen ook het werkvlak achter de tekens laten knipperen, door een extra kleurnummer toe te voegen.
U ziet het werkvlak(PAPER) afwisselend groen en lichtgeel knipperen, indien u intypt:

ink 0,9,24 [ENTER]

Hier is 0 het PAPERnummer en 9 de kleur groen, waarbij de alternatieve kleur lichtgeel is.

RESET nu de computer met **[CTRL],[SHIFT] en [ESC]**.

Let erop dat in mode 0, PEN 14 en 15 respectievelijk PAPER 14 en 15 van twee knipperende kleuren voorzien zijn. Hiervoor is het niet noodzakelijk een extra cijfer aan de INK instructie toe te voegen.
Typ in:

mode 0 [ENTER]
pen 15 [ENTER]

Op het beeldscherm knippert nu het woord READY in hemelsblauw en rose.
Typ in:

paper 14 [ENTER]
cls [ENTER]

U ziet nu dat niet alleen het woord READY knippert, in hemelsblauw en rose, maar ook de achtergrondkleur (PAPER), in geel en blauw.

Deze startkleuren kunnen ook met de INK instructie voor PEN of PAPER worden veranderd. U verandert de kleuren voor PEN in knipperend zwart en wit met de volgende instructie:

ink 15,0,26 [ENTER]

In dit geval is 15 het PEN nummer, 0 is de kleur zwart en 26 is de alternatieve kleur wit.

Tot slot laten we de rand(**BORDER**) knipperen in twee kleuren. We geven in de BORDER instructie twee kleurnummers aan. Typ in:

border 6,9 [ENTER]

De rand knippert nu in lichtrood en groen.

RESET nu de computer met **[CTRL], [SHIFT] en [ESC].**

DEMONSTRATIE PROGRAMMA

Om de andere kleuren, die er zijn, te laten zien typ dan het volgende programma in en laat het RUNnen.
We hebben enkele geluids effecten (**SOUND**) in het programma opgenomen. Deze zullen later worden verklaard:

```
10 mode 0: ink 0,2: ink 1,24: PAPER 0 [ENTER]
20 pen 1: FOR b=0 to 26: border b [ENTER]
30 locate 3,12: print "Randkleur"; b [ENTER]
40 sound 4,(40-b) [ENTER]
50 for t=1 to 600: next t: next b: cls [ENTER]
60 for p=0 to 15: paper p: pen 5: print "Werkvlak"; p: print [ENTER]
70 for n=0 to 15: pen n: print "pen"; n [ENTER]
80 sound 1,(n*20 + p) [ENTER]
90 for t=1 to 100: next t: next n [ENTER]
100 for t=1 to 1000: next t: cls: next p [ENTER]
110 cls: paper 0: pen 1: locate 7,12: print "EINDE": for t=1 to 2000:
    next t [ENTER]
120 mode 1: border 1: ink 0,1: ink 1,24: paper 0: pen 1 [ENTER]

run [ENTER]
```

GRAPHICS

Vanaf nu zullen we niet meer na elke regel [ENTER] aangeven, want we gaan ervan uit dat u dit nu automatisch doet.

De computer heeft een hele reeks symbolen intern opgeslagen. Met de **CHR$()** instructie kunt u deze symbolen afdrukken. Tussen de haakjes hoort het nummer van het symbool te staan: een getal tussen 32 en 255.

RESET nu de computer met **[CTRL], [SHIFT] en [ESC]** en typ dan in:

```
print chr$(250)
```

Op het beeldscherm ziet u symbool 250: een mannetje dat naar rechts loopt. Het volgende programma kunt u gebruiken om alle voorhanden zijnde symbolen met het daarbij behorende nummer af te drukken. Typ het programma in, maar vergeet niet na elke regel de [ENTER]toets in te drukken:

```
10 for n=32 to 255: print n;CHR$(n);
20 next n
run
```

In aanhangsel III vindt u alle symbolen met de daarbij behorende nummers.

LOCATE (Positie v.d. Cursor)

Deze instructie zet de cursor op een gewenste plaats op het beeldscherm. Zonder LOCATE instructie begint de cursor in de bovenste linker hoek van het beeldscherm. Deze heeft de X-,Y-coordinaten 1,1 (X is horizontaal, Y is verticaal). In mode 1 staan 40 posities en 25 regels ter beschikking. De cursor positie in het midden v.d. bovenste regel heeft in mode 1 de X-,Y-coordinaten 20,1. Om dit te zien typt u in (denkt u aan [ENTER]):

```
mode 1   (het beeldscherm wordt gewist, met de cursor linksboven)

10 locate 20,1
20 print chr$(250)
run
```

Om u te overtuigen dat dit de bovenste regel is, typt u in:

```
border 0
```

De rand(BORDER) is nu zwart en het kleine mannetje is op de bovenste regel te zien.

In mode 0 staan maar 20 kolommen ter beschikking, maar nog wel dezelfde 25 regels. Typ in:

```
mode 0
run
```

Het kleine mannetje staat nu in de rechter bovenhoek, want de X-coordinaat 20 is de laatste positie in MODE 0.

In mode 2 zijn 80 posities en 25 regels. We gebruiken hetzelfde programma. U kunt zeker wel raden waar het kleine mannetje nu staat

```
mode 2
run
```

Ga terug naar MODE 1 door in te typen:

```
mode 1
```

Probeer nu zelf, door de waarden van LOCATE en CHR$() te veranderen en daarmee de verschillende symbolen overal op het beeldscherm te plaatsen. Typ bv.:

```
locate 20,12: print chr$(240)
```

U ziet een pijl in het midden v.h. beeldscherm. In deze instructie is:

20	: de horizontale coordinaat(x)	(Bereik 1-40)
12	: de vertikale coordinaat(y)	(Bereik 1-25)
240	: het nummer van het symbool	(Bereik 32-255)

Het symbool 250 kunt u met het volgende programma opnieuw over het hele beeldscherm printen, typ in:

```
5 cls
10 for x = 1 TO 39
20 locate x,20
40 print chr$(250)
50 next x
60 goto 5
run
```

Druk 2 maal op de [ESC]toets om het programma af te breken. Om het vorige teken op het beeldscherm te wissen, voordat het volgende teken geprint wordt, typt u:

```
40 print " "; chr$(250)
```

(Deze nieuwe regel 40 overschrijft de vorige regel 40)

Nu typt u:

```
run
```

Om de beweging van het teken dwars over het beeldscherm te verbeteren, voegt u de volgende regel toe:

```
30 call &bd19
```

Het programma kan nog meer verbeterd worden. Wij voegen enkele vertragings-lussen in en gebruiken ook een ander symbool voor de beweging de andere kant op.
Typ in:

```
list
```

Nu vult u het programma aan met de volgende regels:

```
60 for n=1 to 300: next n
65 for x=39 to 1 step-1
70 locate x,20
75 call &bd19
80 print chr$(251);" "
85 next x
90 for n=1 to 300: next n
95 goto 10
run
```

Probeert u dit interessante programma ook uit. Wij hebben enkele instructies gebruikt die we later zullen uitleggen. Typ nu gewoon in:

```
10 mode 1
20 locate 21,14:print chr$(244)
30 tag
40 for x=0 to 624 step 2
50 mover -16,0
60 if x<308 OR x> 340 then y=196:goto 90
70 if x<324 then y=x-104:goto 85
80 y=536-x
85 sound 1,0,20,7
90 move ox, oy:print" ";:ox=x:oy=y
100 move x,y
110 if (x mod 4) = 0 then print chr$(250);else print chr$(251);
120 for n=1 to 4: call &bd19:next n
130 next x
140 tagoff
150 goto 20
run
```

PLOT

Anders dan de LOCATE instructie gebruiken we de PLOT instructie om de graphics cursor te positioneren, ditmaal met pixel-coordinaten (een pixel is een klein beeldpuntje).

Let erop dat de graphics cursor onzichtbaar is en anders dan de normale cursor. We hebben 640 horizontale pixels en 400 vertikale pixels. De X - Y - coordinaten beginnen linksonder met 0,0.

Deze pixel coordinaten blijven in mode 0, 1 en 2 altijd hetzelfde. Om dit te testen, RESET u de computer met [CTRL], [SHIFT] en [ESC] en typt dan in (denkt u aan [ENTER]):

```
plot 320,200
```

Een klein puntje verschijnt in het midden van het beeldscherm. Nu verandert u de modus met:

```
mode 0
plot 320,200
```

Het puntje is nog steeds in het midden, maar nu groter. Wijzig nog een keer de modus en typ in:

```
mode 2
plot 320,200
```

Het puntje is nog steeds in het midden maar nu veel kleiner. Probeer nog enkele, op verschillende plaatsen op het beeldscherm aan te brengen. Wanneer u klaar bent wist u het beeldscherm en keert u terug in mode 1, door te typen:

```
mode 1
```

DRAW

Reset de computer met [CTRL][SHIFT]en[ESC]

De **DRAW** instructie tekent een lijn uitgaande vanuit de huidige positie van de graphics-cursor. Teken een vierkant op het beeldscherm met het volgende programma. We zetten dan de graphics-cursor met de PLOT instructie op de gewenste plaats, dan tekenen we een lijn van hieraf naar de linker bovenhoek, dan van daar naar de rechter bovenhoek, enz..

Typ in

```
5 cls
10 plot 10,10
20 draw 10,390
30 draw 630,390
40 draw 630,10
50 draw 10,10
60 goto 10
run
```

Druk nu tweemaal op [ESC] om het programma af te breken, voeg nu de volgende regel in het programma, zodat een tweede rechthoek in de eerste wordt getekend.Typ in

```
60 plot 20,20
70 draw 20,380
80 draw 620,380
90 draw 620,20
100 draw 20,20
200 goto 10
run
```

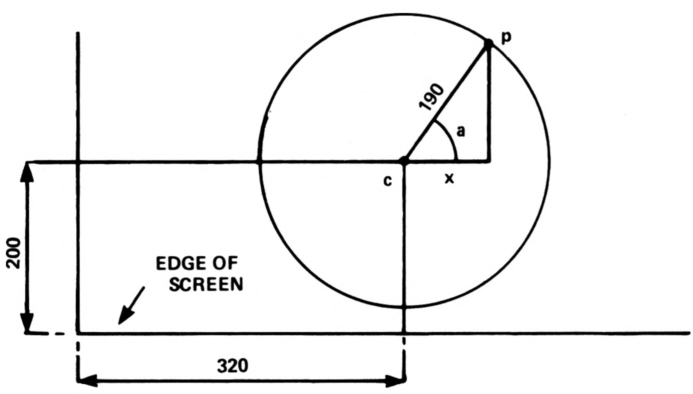
Druk tweemaal [ESC] om het programma af te breken.

CIRKELS

Cirkels kunnen geplot of getekend worden. Men kan een cirkel maken, door de x,y-coordinaten van elk puntje op de cirkelomtrek te plotten.

Bekijk de tekening hieronder en u zult ontdekken dat punt p op de cirkelomtrek met de x en y coordinaten kan worden bepaald.

x = 190*cos(a) en **y =190*sin(a)**



NEW

De BASIC instructie NEW hebben we reeds gebruikt, voordat we het eigenlijke programma intypten. NEW zegt de computer dat het geheugen gewist moet worden ongeveer zoals we de [CTRL][SHIFT]en [ESC] zouden gebruiken. Het beeldscherm wordt echter niet gewist, alleen het intern geheugen. Het is soms nuttig het oude programma op het scherm te hebben, terwijl we het nieuwe programma schrijven.

Hoe men de punten van een cirkel plot

In het vorige programma hebben we alle punten vanuit de onderste linker hoek v.h. beeldscherm geplot. We willen nu de cirkel in het midden van het beeldscherm hebben, dus op de coordinaten 320,200, zodat we de coordinaten bij onze berekende punten optellen moeten.

Een programma voor een cirkel kan er als volgt uitzien. Typ in :

```
new
5 cls
10 for a=1 to 360
15 deg
20 plot 320,200
30 plot 320+190*cos(a), 200+190*sin(a)
40 next
run
```

De straal van de cirkel kan verkleind worden door het getal 190 te verkleinen.(190 bepaalt de pixels).

De cirkel wordt anders gePLOT, wanneer we regel 15 verwijderen. Wis regel 15 door eenvoudig

```
15
```

in te typen.

Een schijf wordt getekend, wanneer we de instructie PLOT door DRAW vervangen. Regel 30 zou dan zijn:

```
draw 320+190*cos(a), 200+190*sin(a)
```

Probeer het eens uit met en zonder regel 15.

We hebben in regel 40 gezien dat we **next** inplaats van **next a** schreven.

Het is toegestaan, alleen next te schrijven. De computer zoekt uit, welke FOR bij welke NEXT hoort. Voor programma´s waar veel FOR en NEXT in staat, is het overzichtelijker wanneer u de namen van de variabelen toevoegt aan NEXT.

ORIGIN

In het vorige programma hebben we de PLOT instructie gebruikt om het midden van de cirkel te bepalen en hebben dan de x, en y coordinaten bij deze positie opgeteld. We kunnen ook werken met ORIGIN. Hiermee bepalen we het midden van de cirkel, zodat we alleen alle punten van de omtrek van de cirkel moeten bereken. Typ het volgende programma in om dit te zien:

```
new
5 cls
10 for a=1 to 360
15 deg
20 origin 320,200
30 plot 190*cos(a),190*sin(a)
40 next
run
```

U kunt regel 15 verwijderen of in regel 30 PLOT in DRAW veranderen, om een schijf vanuit het middelpunt te tekenen.

Het volgende programma maakt 4 kleine cirkels,typ in:

```
new
5 cls
10 for a=1 to 360
15 deg
20 origin 196,282
30 plot 50*cos(a),50*sin(a)
40 origin 442,282
50 plot 50*cos(a),50*sin(a)
60 origin 196,116
70 plot 50*cos(a),50*sin(a)
80 origin 442,116
90 plot 50*cos(a),50*sin(a)
100 next
run
```

U kunt ook hier regel 15 verwijderen en in regel 30,50,70 en 90 de DRAW instructie gebruiken.

GOSUB RETURN

Een serie van instructies welke in een programma op verschillende plaatsen nodig zijn, kunnen in een subprogramma (zgn. subroutine) ondergebracht worden. Deze subroutine kan met GOSUB gevolgd door het regelnummer worden aangeroepen.

U eindigt een GOSUB routine met RETURN. Vanaf die plaats keert de computer terug naar de regel welke op de laatst opgevolgde GOSUB volgt.

In het vorige programma werd het statement: PLOT 50*cos(a), 50*sin(a) vier maal geschreven. Dit statement kan als subroutine geschreven worden en met GOSUB opgeroepen worden. Typ het volgende:

```
5 cls
10 for a=1 to 360
15 deg
20 origin 196,282
30 gosub 120
40 origin 442,282
50 gosub 120
60 origin 196,116
70 gosub 120
80 origin 442,116
90 gosub 120
100 next
110 end
120 plot 50*cos(a),50*sin(a)
130 return
run
```

We hebben END in regel 110 toegevoegd, zodat het programma niet automatisch na regel 100 doorloopt naar regel 120, welke alleen maar voor de GOSUB routine wordt gebruikt.

We sluiten dit onderdeel af met het volgende programma waarin veel BASICinstructies voorkomen. Deze zult u nu hopelijk begrijpen. Typ in:

```
new
10 mode 0: border 6: paper 0: ink 0,0
20 gosub 160: for x=1 to 19: locate x,3
30 pen 15: print "  "; chr$(238)
40 for t=1 to 50: next t: sound 2,(x+100)
50 next x: gosub 160: for b=3 to 22
60 locate 20,b: pen 7: print chr$(252)
70 cls: gosub 160: next b
80 sound 2,0,100,15,0,0,1
90 gosub 160: border 16,24: locate 20,25
100 pen 14: print chr$(253);
110 for t=1 to 1000:next t
120 border 6: gosub 160: for f=3 to 24
130 locate 10,(25-f): pen 2
140 print chr$(144): cls: gosub 160
150 sound 7,(100-f),5: next f: goto 10
160 locate 10,25: pen 12
170 print chr$(239): return
run
```

SOUND

Geluids-effecten worden via een interne luidspreker weergegeven. Wanneer u met de MP1 modulator/voeding werkt, moet u het geluid van het TV toestel afzetten.

De geluidssterkte in de computer kan met de VOLUME-regelaar aan de rechterkant van de computer worden ingesteld. U kunt geluid ook op de **AUX**iliary-ingang van uw stereo-installatie aansluiten door de (I/O) uitgang aan de achterkant van uw computer te gebruiken. U kunt de muziek dan in stereo via uw stereo-installatie horen.

De SOUND instructie heeft 7 parameters; de eerste 2 moet u aangeven, de resterende naar keuze. De instructie ziet er als volgt uit:

SOUND kanaal-status, toonhoogte, lengte, geluidssterkte, sterkte-verloop, toonhoogteverloop, ruisfrequentie.

In de volgende voorbeelden bepalen we de kanaal-status, via referentie nummer 1, op kanaal 1.

toonhoogte (tone period)

In aanhangsel VII ziet u dat de middelste C de toonhoogte 478 heeft. Typ in:

```
10 sound 1,478
run
```

en u hoort hoe de **C** 0,2 seconde lang klinkt.

lengte (duration)

Wanneer u de lengte niet aangeeft, wordt een lengte van 0,2 seconde aangenomen. De eenheid voor de lengte is 0,01 seconde. Om een toon 1 sec. lang te laten klinken moet u 100 aangeven; voor 2 sec. 200, enz.

```
10 sound 1,478,200
run
```

De C klinkt 2 sec. lang.

geluidssterkte (volume)

Hiermee geeft u de aanvangsgeluidssterkte van een noot aan. U kunt waarden van 0 tot 7 aangeven. Wanneer u echter een geluidssterkte-verloop aangeeft is dit bereik 0 tot 15. Indien u geen waarde intypt, wordt 4 aangenomen.

```
10 sound 1,478,200,3
run
```

Let op de geluidssterkte. Geef nu een hogere waarde:

```
10 sound 1,478,200,7
run
```

U hoort nu dat deze toon veel harder is.

geluidssterkte variatie (volume envelope)

De instructie voor de geluidssterkte variatie heet **env (ENvelope Volume)** Deze heeft normaal 4 parameters, waarvan de laatste 3 in elk van de 5 mogelijke variatiegebieden kunnen voorkomen. Wij gebruiken hier maar 1 deel. Verdere verklaringen vindt u in hoofdstuk 6.

ENV variatiewaarde, aantal stappen, stap-amplitude(grootte), stap-tijd

variatiewaarde (envelope number)

Deze waarde wordt in de SOUND instructie aangegeven, om de betreffende variatie te krijgen. Het bereik van de variatie ligt tussen 0 en 15

aantal stappen (number of steps)

Deze parameter wordt in samenhang met de stap-tijd gebruikt. U wilt bijv. 10 stappen van elk een seconde hebben, in dit geval is het aantal stappen 10. Het stappengebied loopt van 0 tot 127.

stap amplitude (amplitude of step)

Elke stap kan in amplitude (sterkte) varieren van 0 tot 15, met betrekking tot de laatste stap. Deze 15 geluidssterktes zijn hetzelfde als in de SOUND instructie. De stap kan echter van -128 tot + 127 varieren, zodat de amplitude niet alleen normaal omhoog en omlaag te veranderen is, maar ook hogere dan 15, waardoor we bijzondere effecten krijgen. De stap amplitude kan waarden van -128 tot +127 hebben.

stap tijd (step time)

Dit getal bepaalt de tijd tussen de stappen en wordt in eenheden van 0,01 seconden gemeten. Het bereik loopt van 0 tot 255, zodat de langste tijd tussen de stappen 2,55 seconden bedraagt.

Om met de geluidssterkte variatie te oefenen, typen we het volgende programma in

```
5 env 1,10,1,100
10 sound 1,284,1000,1,1
run
```

Regel 10 definieert een toon met een toonhoogte van 284 (standaard-toon a), die 10 seconden duurt, een aanvangsgeluidssterkte van 1 heeft en de geluidssterkte variatie 1 gebruikt. Regel 5 definieert de variatie 1 in 10 stappen, waarbij de geluidssterkte elke seconde met 1 verhoogt wordt (100x0,01 seconde)
Verander regel 5 telkens als volgt en hoor de verschillende effecten van elke variatie:

```
5 env 1,100,1,10
5 env 1,100,2,10
5 env 1,100,4,10
5 env 1,50,20,20
5 env 1,50,2,20
5 env 1,50,15,30
```

Probeer tot slot dit uit:

```
5 env 1,50,2,10
```

U merkt hierbij dat de geluidssterkte na de helft van de tijd constant blijft. Dat is gebeurd omdat we 50 stappen hebben die elk 0,1 seconde lang waren.De totale tijd voor de geluidssterkte verandering bedroeg 5 seconden, de tijd voor de toon in de SOUND instructie is 10 seconden (getal 1000)

Probeer nu zelf, welke geluiden u kunt maken.

toonhoogte variatie (tone envelope)

De instructie voor de toonhoogte variatie heet **ent (ENvelope Tone)**. De instructie ENT heeft normalerwijze 4 parameters. De laatste 3 kunnen in elk van de 5 mogelijke variatiegebieden voorkomen. Wij gebruiken hier maar 1 deel. Verdere verklaringen vindt u in hoofdstuk 6

ent variatiewaarde, aantal stappen, toonhoogte van de stap, stap tijd.

variatiewaarde (envelope number)

Deze waarde wordt aan een bepaalde variatie toegekend, zodat het kan worden gespecificeerd in de SOUND instructie. Het variatiegebied gaat van 0 tot 15

aantal stappen (number of steps)

Deze parameter wordt in samenhang met de stap tijd gebruikt. U kunt bijv. 10 stappen van elk 1 seconde willen. Toegestaan zijn 0 tot 239 stappen.

toonhoogte per stap (tone period of step)

De toonhoogte mag van -128 tot +127 varieren. Negatieve stappen verhogen de frequentie van de tonen. De hoogste toon is 0. Daaraan moet u denken, wanneer u de toonhoogte variatie berekent. Aanhangsel VII geeft een overzicht van alle mogelijke toonhoogtes. De toonhoogte stappen kunnen waarden tussen -128 tot +127 aannemen.

STAP TIJD (step time)

Dit getal bepaalt de tijd tussen de stappen en wordt in eenheden van 0,01 seconden gemeten. Het waardebereik loopt van 0 tot 255, zodat de langste tijd tussen de stappen 2,55 seconde bedraagt. Om met de toon-variatie te oefenen typen we het volgende in:

```
5 ent 1,100,2,2
10 sound 1,284,200,15,0,1
run
```

Regel 10 specificeert een toon met een hoogte van 284 (standaard-toon a) die 2 sec., duurt een aanvangsgeluids-sterkte van 15 (maximale waarde) heeft en de toon-variatie nummer 1 gebruikt.

Regel 5 specificeert de toon-variatie 1 bestaande uit 100 stappen, waarbij de toonhoogte elke 0,02 seconden met 2 toeneemt (de frequentie wordt daardoor steeds lager).

Verander regel 5 telkens als volgt en laat het programma iedere keer lopen (RUN). Luister naar het effekt van de wijziging van de toon-variatie :

```
5 ent 1,100,-2,2
5 ent 1,10,4,20
5 ent 1,10,-4,20
```

Vervang nu de SOUND instructie en de toon-variatie door:

```
5 ent 1,2,17,70
10 sound 1,142,140,15,0,1
15 goto 5
run
```

Druk tweemaal op **[ESC]** om het programma te stoppen.

U kunt nu de geluidssterkte-variatie, de toon-variatie en de SOUND instructie combineren om zo verschillende klanken te verkrijgen. Begin met het volgende voorbeeld:

```
new
5 env 1,100,1,3
10 ent 1,100,5,3
20 sound 1,284,300,1,1,1
run
```

Vervang dan regel 10 door:

```
10 ent 1,100,-2,3
```

Vervang alle regels als volgt:

```
5 env 1,100,2,2
10 ent 1,100,-2,2
20 sound 1,284,200,1,1,1
run
```

Probeer ook uw eigen variaties, <u>veel plezier !</u>

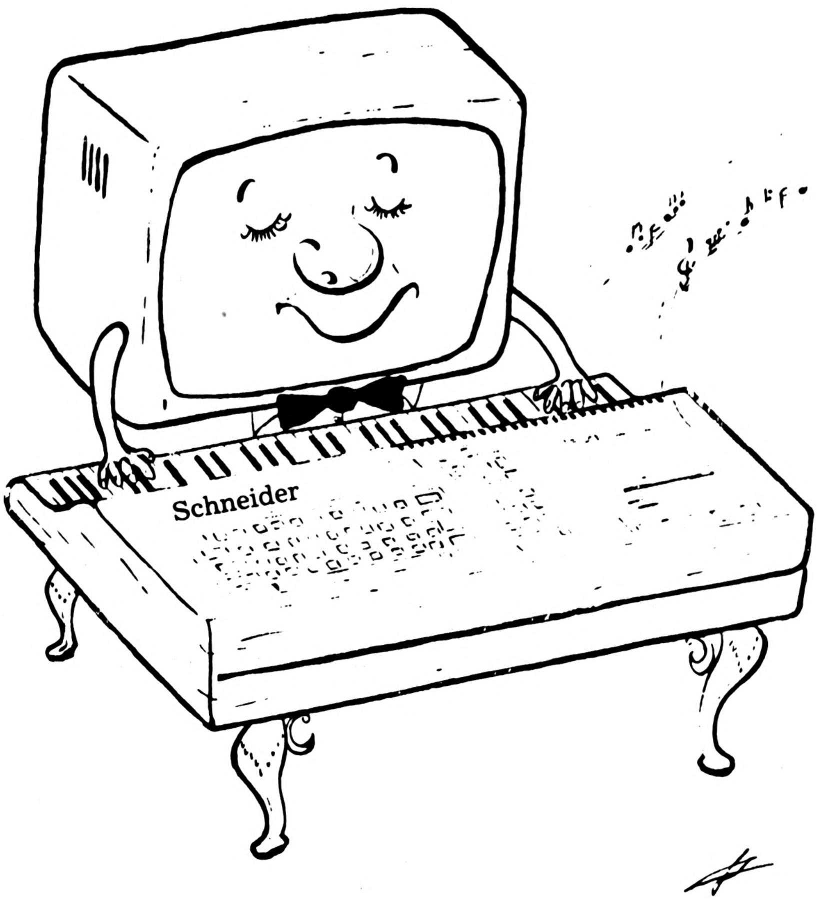
ruis (noise)

U kunt aan het einde van de SOUND instructie ook ruis toevoegen. Er is een ruiswaarde in een bereik van 1 tot 31. Probeer dit uit door een ruisgetal aan het einde van de SOUND instructie toe te voegen, achter de ENV instructie (geluidssterkte-variatie).

Vervang de regels 5 en 20 door:

```
5 env 1,100,3,1
20 sound 1,200,100,1,1,1,5
run
```

Probeer nu nog ander ruis te produceren. Verander de geluidssterkte-variatie (ENV) en de SOUND instructie met en zonder ruis.



1 HET BEGIN:

Voor wie de inleiding voor beginners heeft overgeslagen en nu moeilijkheden heeft, met het hierbesprokene, kan beter de inleiding nogmaals doornemen, waarin de opbouw, inschakelen en omgang met het toetsenbord beschreven werd

Thema´s van dit hoofdstuk zijn:

* **begrippen in dit boek**
* **Inschakelen**
* **Vertrouwd raken met het toetsenbord**

Ook wanneer u al ervaring met computers heeft, let dan toch op de volgende regels. Waarschijnlijk heeft u pas het pakket geopend en kunt u niet meer wachten om met de computer te werken. In dit hoofdstuk komt u alles te weten, wat u weten moet om met BASIC te kunnen beginnen. De nieuweling kan echter beter met de inleiding beginnen.

BELANGRIJK dit MOET u lezen.

Terminologie.

Om te kunnen onderscheiden, of toetsen op het toetsenbord of tekst in programma´s worden bedoeld, gelden vanaf hier de volgende regels:

[ENTER]: Toetsen welke bij hun gebruik geen teken op het beeldscherm vertonen, staan in **[]**.

QWERTYUIOP: Toetsen welke bij hun gebruik een teken op het beeldscherm brengen, zijn zo of in normale hoofdletters geschreven, echter zonder haakjes.

10 for n=1 to 1000: Tekst die op het beeldscherm verschijnt of via het toetsenbord ingegeven wordt, is in die vorm afgedrukt. Let op het verschil tussen de 0 (nul) en de hoofdletter O.

Er wordt verondersteld dat u elk programma of ieder direkt bevel door het drukken van de **[ENTER]** toets af sluit. Deze aanwijzing wordt in de rest van het boek niet meer gegeven.

Verder wordt verondersteld dat u **run** intypt wanneer een programma is ingetikt. BASIC verandert elke kleine letter in hoofdletter, na **LIST**, wanneer het een BASIC-instructie betreft. De voorbeelden van hier af aan worden met hoofdletters geschreven, omdat ze ook zo uitge**LIST** worden. Wanneer u echter BASIC instructies in kleine letters ingeeft, kunt u fouten in zulke woorden gemakkelijk herkennen, omdat foutieve BASIC instructies met **LIST** niet veranderd worden, en dan in kleine letters blijven.

1.1.1 OPEN DE DOOS

Het kleurenbeeldscherm

BELANGRIJK !

Lees aub. de installatiebeschrijving aan het begin van dit handboek

Als de computer, zoals in tekening 1 is aangegeven, is aangesloten, schakel dan het beeldscherm in en pas dan de computer met de schuifschakelaar aan de rechterzijde. Na enige (opwarm) tijd verschijnt het volgende beeld op het scherm:

```
SCHNEIDER 64K Microcomputer  (v1)

©1984 Amstrad Consumer Electronics plc
          and Locomotive Software Ltd.

BASIC 1.0

Ready
█
```

Dit beeld komt steeds als de computer ingeschakeld wordt of als men de volgende toetsen in de aangegeven volgorde indrukt **[CTRL] [SHIFT] [ESC]**. Door deze toetsen tegelijk in te drukken komt de computer in zijn begin toestand. Oefen nu de koude start met deze drie toetsen.

Stel nu de helderheid in met BRIGHTNESS, (rechts onder aan de zijkant van het beeldscherm), de kleur is al ingesteld. Wanneer u de kleur wilt veranderen (geel schrift op blauwe achter grond) moet u dit per programma apart doen. Zie graphics aanwijzing in hoofdstuk 8.

Wanneer het u niets uitmaakt om op de leerstof voor te lopen, dan vindt u hier een kort programma, welke de tekst, in de 80 teken mode, in de best leesbare kleurencombinatie op het scherm zet.

Voer in:

```
 10 mode 2
 20 ink 1,0
 30 ink 0,13
 40 border 13
```

U kunt dit programma ook in de direkt-modus(directmode) invoeren.

Wanneer u dit programma niet begrijpt of niet kunt intypen dan moet u terug gaan naar de inleiding voor beginners. De 80 teken/regel-mode is de comfortabelste voor het ontwikkelen van programma's. U kunt het programma, indien u hoofdstuk 2 heeft gelezen, aan het begin van een lege cassette SAVEn, zodat u de invoer kunt sparen bij herhalingen.

Let er wel op dat u niet te dicht bij het beeldscherm gaat zitten of de helderheid te hoog instelt, u kunt last krijgen van uw ogen. Wanneer dat het geval is, schakel dan uit en ga iets anders doen.

1. zorg voor goed licht tijdens het lezen van dit handboek.
2. stel de helderheid van het beeldscherm zo laag mogelijk in.
3. ga ver genoeg van het beeldscherm vandaan zitten.

1.1.2 De groene monitor

Het GT 64 beeldscherm heeft 3 regelaars. Helderheid, contrast en vertikale synchronisatie kunnen hiermee geregeld worden. De vertikale synchronisatie hoeft normaal, maar eenmaal ingesteld te worden, helderheid en contrast hangen af van de omstandigheden waaronder de CPC464 gebruikt wordt.

Gebruikt men een monochroom beeldscherm dan verschijnt na het inschakelen dezelfde melding als bij een kleurenmonitor, maar wel in lichtgroen op een donkere achtergrond. De veschillende kleuren worden bij een monochroom beeldscherm door helderheidsverschil ten opzichte van de achtergrondkleur getoond.

Hoewel veel werken aan het beeldscherm, vermoeide ogen kan veroorzaken , is een monochroom beeldscherm beter voor de ogen dan een kleuren-beeldscherm, omdat de oplossing (aantal beeldpuntjes) bij een monochroom beeldscherm beter is. Het beeldscherm is stabiel en knippert niet. Regel helderheid en contrast tot een goed beeld ontstaat.

Om het beeldscherm in de 80 tekens-mode te brengen, typen we het volgende programma in en SAVEn het op een cassette:

```
 10 REM stel beeldopbouw in
 20 FOR n=0 TO 26
 30 MODE 2
 40 INK 1,n
 50 INK 0,(26-n)
 55 BORDER n
 60 LOCATE 15,12: PRINT"Druk een toets om het beeldformaat te
    veranderen
 70 a$=INKEY$
 80 IF a$="" GOTO 70
 90 NEXT
 100 GOTO 20
```

Het programma hierboven laat een verdere bijzonderheid zien, welke dit handboek betreft.

Soms wordt er bij LISTen van een programma, een gedeelte van een programma-regel op de volgende schermregel gezet. spaties aan het eind van een regel dienen daarbij alleen maar voor betere leesbaarheid en bezetten in een programma geen geheugenplaatsen. Daarmee zijn nog niet alle mogelijkheden van het printen op het scherm uitgeput, maar u heeft een indruk gekregen. Wanneer een combinatie u goed bevalt, druk dan 2 maal op [ESC] en het programma blijft staan. ***BREAK*** wordt geprint.

In volgende beschrijvingen zullen we meer de aandacht vestigen op de beeldscherm uitwerking. Programma's met interessante kleur- en graphics-mogelijkheden kunnen op een monochroom scherm minder goed uit de verf komen, ofschoon er veel aandacht werd besteed aan een uitgewogen reeks grijsniveau's (zie ook hoofdstuk 5).

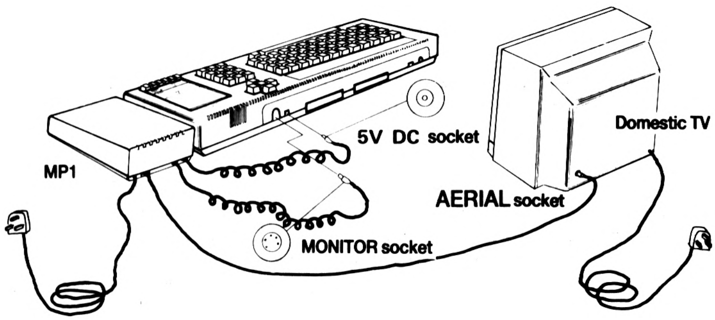
Het voordeel van het groene beeldscherm is de helderheid, en precieze uitbeelding, speciaal bij het ontwerpen van programma's. Heeft u echter de smaak te pakken, dan zult u met minder dan kleur geen genoegen willen nemen.

1.1.3 De MP1 modulator/voeding

De MP1 modulator/voeding is een randapparaat dat u kunt kopen om in plaats van de groene GT64 monochroom beeldscherm, uw normale kleuren TV te gebruiken.

BELANGRIJK !

Lees eerst de instalatiebeschrijving aan het begin van dit boek.



Afbeelding 2: Het aansluiten van de MP1, computer en het TV toestel.

Het MP1 randapparaat moet rechts van het toetsenbord worden geplaatst, zoals in afb. 1.
Zet nu de geluidssterkte van het TV toestel zo laag mogelijk (de CPC 464 heeft zijn eigen luidspreker). Het geruis dat u uit uw TV-toestel hoort is normaal. Schakel eerst het TV toestel in en dan de computer met de schuifschakelaar aan de rechter zijde (POWER).

Het rode lampje (in Function) rechts op het toetsenbord moet oplichten, u moet nu de TV instellen om het signaal van de computer te ontvangen.

Wanneer u een TV toestel heeft waar u de kanalen via druktoetsen moet kiezen, neem dan een vrije toets of een ongebruikt kanaal. Stel de kanalenkiezer op ongeveer kanaal 36, en stem af tot u een beeld krijgt, dat er uitziet als het bekende startbeeld.

Regel de TV zo dat de beste beeldscherpte bereikt wordt. De tekst is goudgeel op diepblauwe achtergrond, dit kan per TV iets afwijken.

Wanneer uw TV een draaiknop voor programmakeuze heeft, draai deze dan zolang tot het beeld zuiver en rustig verschijnt (ook ± kanaal 36).

Omdat het signaal eerst wordt gemoduleerd en dan weer wordt gedemoduleerd, kan de kwaliteit van het TV signaal hieronder lijden. Het beeld zal niet zo goed zijn als bij een video monitor en afhankelijk van de kwaliteit van uw eigen TV toestel, kan het zijn dat de 80 karakters mode (mode 2) geen bevredigend resultaat geeft. In dat geval kunt u het beste mode 1 gebruiken.

1.2 DE EERSTE STAPPEN

De apparaten zijn aangesloten: de monitor of TV aangesloten, netstekker in het stopcontact. De computer is ingeschakeld en wacht nu op uw invoer.

Het startbeeld is de enige "ingebouwde" tekst die kan verschijnen zonder dat u een enkele instructie heeft moeten invoeren. Wanneer u reeds BASIC kent, heeft u misschien al een klein programma ingevoerd om zodoende met het systeem vertrouwd te raken.
Schneider BASIC zal u misschien in een aantal opzichten bekend voorkomen.

1.2.1

Om u een beetje wegwijs te maken laten wij u een klein programma zien dat de ingebouwde tekens van de computer laat zien. Dat is de tekenset die via het toetsenbord ingevoerd, op het beeldscherm of de printer kan worden afgedrukt.

Enige van de tekens, die u ziet kunnen niet direkt via het toetsenbord worden ingevoerd, maar alleen met behulp van de **PRINT CHR$**-(getal) instructie afgedrukt worden, hetgeen later beschreven wordt.

Dit komt doordat alle informatie in een computer in zogenaamde bytes opgeslagen wordt. In aanhangsel 2 kunt u zien dat een byte 256 verschillende combinaties kan bevatten. Er kunnen dus 256 verschillende tekens gemaakt worden, waarvan echter op de normale toetsenborden en printers maar ca. 96 gebruikt worden.

De tekenset van de computer is een onderdeel van een standaardset. Deze wordt in de computerwereld, ASCII systeem genoemd, wat een afkorting is van:

American
Standard
Code for
Information
Interchange

Het is een systeem waarmee data van de ene computer naar een andere kan worden overgedragen. Het is zo ongeveer het enige in de computerwereld dat universeel is. Wij raden u daarom aan zich met de ASCII tekens vertrouwd te maken. Aanhangsel III toont de ASCII tekenset en alle verdere tekens die er op de CPC464 zijn, met hun numerieke waarden.

Enkele van de niet afdrukbare tekens kunnen toch worden afgedrukt, wanneer u de **CONTROL ([CTRL])** toets te samen met een andere toets indrukt. Houdt u zich hier echter niet mee bezig totdat u de betekenis van die toetsencombinaties kent.

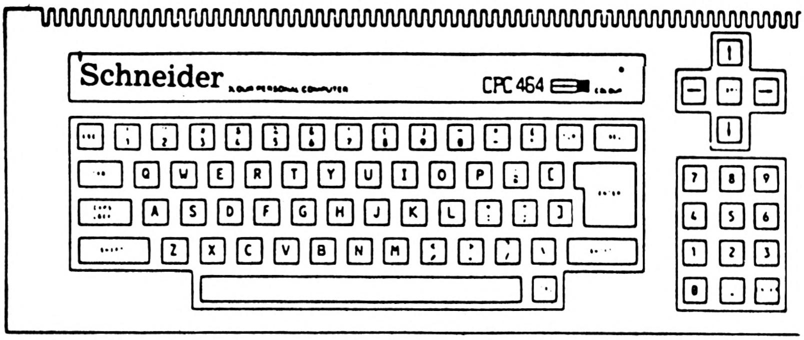
1.2.2

Opdat u precies ziet hoe deze tekens er uitzien, typt u het volgende programma in. Wij zullen uw nieuwsgierigheid naar de CPC464 testen, daarbij ziet u ook hoe eenvoudig programma's gemaakt worden en dat er geen problemen met de hardware zijn, zolang u bij inschakelen steeds weer het startbeeld krijgt. De CPC 464 wacht eenvoudig erop door u op de juiste wijze geprogrammeerd te worden.

(Wanneer u bij het invoeren van een programma een fout maakt, kijk dan bij 1.2.7 want daar ziet u hoe correcties gemaakt worden, zonder het programma opnieuw te moeten invoeren).

Hoofd en kleine letters spelen geen rol bij het programmeren. De woorden moeten alleen door SPATIES, KOMMA'S, PUNT-KOMMA'S of andere tekens worden gescheiden, die bestemd zijn voor het scheiden van tekst, omdat Schneider BASIC het gebruik van gereserveerde woorden, die als BASIC instructies gebruikt worden, in variabelen toestaat (zie aanhangsel VIII).

Het toetsenbord wordt in afb.4 getoond, het gaat hier om het originele toetsenbord. Veel van de toetsen kunnen van andere waarden worden voorzien, zoals in het volgende hoofdstuk beschreven wordt.



Afb.4: Het toetsenbord van de CPC464

Wanneer u **[ENTER]** drukt wordt de programmaregel of instructieregel afgesloten en ingevoerd. Zonder regelnummer wordt deze invoer, direkt verwerkt(DIRECT MODE), in plaats van als programmaregel te worden opgeslagen.

[ENTER] wordt soms ook **Carriage Return** of eenvoudig return genoemd. Deze benoeming stamt nog uit de tijd, toen computerterminals als mechanische schrijfmachines opgebouwd waren. De uitdrukking is gebleven en zo wordt in de ASCII tekenset de code voor **[ENTER]** met de letters ´CR´ aangeduid. Geef eens in:

```
10 FOR N = 32 TO 255
20 PRINT CHR$(N)
30 NEXT N
RUN
```

1.2.3

Kijk wat er op het beeldscherm verschijnt:

```
Schneider 64K Microcomputer  (v1)

©1984 Amstrad Consumer Electronics plc
        and Locomotive Software Ltd.

BASIC 1.0

Ready
10 FOR N = 32 to 255
20 PRINT CHR$(N);
30 NEXT N
run
```



```
Ready
█
```

De computer werd door dit kleine programma gevraagd alle tekens te printen. Is dat niet het geval, dan heeft u een fout gemaakt. Kijk bij 1.2.7 of u de fout kunt ontdekken.

Wanneer nu alles goed is en het gewenste resultaat heeft, zullen we gaan bekijken wat er gebeurd is.

U heeft de computer niet de opdracht **PRINT "abcdefghijklmn enz"** gegeven, maar u vroeg hem uit te voeren:

```
PRINT CHR$(N)
```

N is een korte naam voor een variabele. U had ook een andere geldige variabelenaam kunnen gebruiken, maar wiskundigen gebruiken nu eenmaal graag deze letter. Een variabele is een opslaggebiedje in het geheugen van de computer, waarvan de waarde via programmainstructies kan worden veranderd.

Een getal als 5 is een vast getal dat tussen 4 en 6 ligt, het is dus geen variabele. N is een vaste letter van het alfabet.
Hoe weet de computer nu het verschil ?

Had u "N" ingetypt dan zou dit als vast gegeven herkend worden en zou met **Syntax error** op zijn gereageerd, omdat FOR "N" niet kan.

Omdat N zonder aanhalingstekens werd gebruikt, weet de computer dat N de naam van een variabele is. De aanwijzing FOR moet worden gevolgd door een variabele en accepteert dan ook de volgende lettercombinatie als variabele, wanneer ze aan de regels beantwoordt.

De uitdrukking **N = 32 TO 255** zorgt dat het bereik voor de variabele N loopt van 32 tot 255. Nadat de variabele gedefinieerd is, moet aan de computer gezegd worden wat ermee gebeuren moet, zoals in de volgende regel:

20 PRINT CHR$(N);

Dit statement zorgt er voor dat de functie CHR$ met de momentele waarde van N het daarmee overeenkomende teken PRINT.

De puntkomma aan het einde van de regel verhindert, dat na elk teken, een nieuwe regel wordt begonnen. Daarmee worden in een regel alle tekens achter elkaar gezet.

Het volgende statement geeft de computer opdracht dat hij na het volbrengen van de vorige opdracht met de eerste waarde uit de reeks (32) voor de variabele N, naar de aanwijzing FOR terug moet gaan. Dan moet de opdracht met de volgende (NEXT) hogere waarde van de variabele N uitgevoerd worden.

Deze constructie noemt men een lus (LOOP), een zeer belangrijke techniek bij het programmeren. Lussen komen voor in elke programmeeertaal. Zij verkorten veel programmma´s. Wanneer u de voordelen kent zult u ze zeker vaak gebruiken.

Wanneer deze FOR-lus zijn grenswaarde (255) bereikt heeft, wordt met de opdracht verder gegaan welke op NEXT volgt. Hier is er echter geen volgende opdracht, daarom stopt het programma de computer en meldt: **Ready.**

Nu kunt u verdere opdrachten invoeren of het zelfde programma nogmaals laten lopen (RUN). Het programma blijft zolang in het geheugen totdat u in de computer een ander programma voert, hem zegt dit programma te vergeten of hem uitschakelt. Wanneer u het programma niet op cassette geSAVEd heeft, is het verloren.

Dit voorbeeld van een programma toont u de elementaire eigenschappen van een computer. Alles wat de computer intern doet berust op getallen. De computer heeft zijn tekenset afgedrukt waarbij hij steeds een getal voor het gewenste teken gebruikt. Wanneer u bijv. toets **A** drukt herkent de computer aan de hand van de waarde van de toets in zijn tabellen, dat deze toets een **A** voorstelt en print het teken op het scherm. Elk teken heeft zijn eigen interne waarde, deze kunt u in aanhangsel III terug vinden.

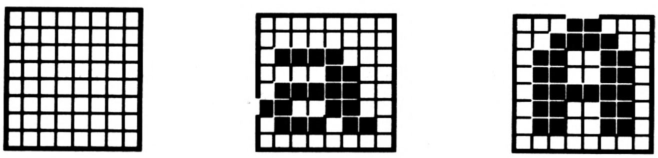
1.2.4

Twijfel niet wanneer u niet alle uitdrukkingen en technische woorden op deze bladzijde begrijpt. De technisch geinteresseerden zal het helpen begrijpen, wat de computer met uw opdrachten doet en hoe u tot het gewenste resultaat komt. Wanneer u dit hoofdstuk te moeilijk vindt, ga dan verder naar 1.2.5.

De waarde voor de letter A is 65. De computer begrijpt 65 ook niet, maar verwerkt dit getal in zogenaamde machinecode. aanhangsel II toont de machinecode.

Het omzetten van de decimale getallen uit het dagelijks gebruik in machinecode (hexadecimale getallen) schijnt in eerste instantie moeilijk te begrijpen zijn, omdat wij aan ons eigen getallensysteem gewend zijn. Hebt u door, hoe HEXadecimale getallen zijn opgebouwd, dan worden veel dingen makkelijker te begrijpen.

De computer verwerkt een ingetypte A in een getallenvorm welke hij begrijpt en zoekt daarmee in een ander deel van zijn geheugen naar een getallen tabel, waarin dat teken is gedefinieerd. Het teken dat men op het beeldscherm ziet wordt door een datablok gemaakt, dat als een numerieke matrix in het geheugen zit.



Tekening 5 Raster voor een teken, kleine letter a, hoofdletter A.

De elementen van deze matrix zijn punten, verdeeld in rijen en kolommen. Het teken wordt gemaakt door in het raster van punten, bepaalde punten "aan" en andere "uit" te zetten. Dit patroon is in het geheugen opgeslagen. 8 regels van 8 hokjes vormen bij de CPC464 een teken (een BYTE). Wanneer u een teken zelf wilt maken, kunt u dat met de instructies **SYMBOL** en **SYMBOL AFTER** (beschreven in hoofdstuk 8).

Zelf gedefinieerde tekens kunnen uit 0 tot 64 punten samengesteld worden; bedenk maar eens hoeveel mogelijkheden dat zijn. Daarbij kunt u patronen aan elkaar koppelen om zo een groot teken te krijgen. U ziet dat het aantal mogelijkheden alleen wordt begrensd door uw tijd en fantasie.

1.2.5 Terug naar het programma !

De uitkomst van het eerste programma, dat u heeft ingevoerd, is nogal slordig. Het startbericht is nog steeds te zien. Het is beter wanneer u het beeld wist, voordat u iets uitvoert. Met een extra instructie kan dit gebeuren.

Typ het volgende in, in de regel waar de cursor staat. Wanneer u niet weet wat een cursor is (het vierkante blokje dat de plaats aangeeft waar het volgende teken komt te staan), dan kunt u beter eerst de inleiding doorlezen!.

```
5 CLS
RUN
```

Nu wordt het beeld gewist, voordat de nieuwe instructie verschijnt.

Dit is een ander belangrijk punt in BASIC. De volgorde waarin de opdrachten worden ingetikt is onbelangrijk. Het programma wordt in de volgorde van de regelnummers uitgevoerd, ook wanneer de programma-LISTing niet op het beeldscherm in de juiste volgorde zichtbaar is. Nieuwe opdrachten worden op de goede plaats gezet en in het geheugen opgeslagen.

1.2.6 LISTing

U kunt steeds met behulp van de instructie LIST het programma in het geheugen op het beeldscherm afdrukken. Voer in:

```
LIST
```

het resultaat is:

```
5 CLS
10 FOR N = 32 TO 255
20 PRINT CHR$(N);
30 NEXT N
Ready
```

Het programma blijft in het geheugen totdat u:
* uitschakelt
* RESET - de toetsen **[CTRL] [SHIFT] [ESC]** na elkaar indrukt.
* een programma van een cassette laadt (LOAD of RUN)
* de instructie **NEW [ENTER]** intikt, waardoor het geheugen wordt gewist (de mode en kleur blijven ongewijzigd).

Nu gaan we een functie in een functie-toets vastleggen, die de volgende functie krijgt: **[ENTER]CLS:LIST[ENTER]**. Hiermee kan men de programma-ontwikkeling enorm versnellen. Voer in:

KEY 138,CHR$(13)+"CLS:LIST"+CHR$(13)

Druk nu op de punt in het numerieke toetsenbord, rechts naast het gewone toetsenbord. Er zijn 32 toetsen die op deze manier voorgeprogrammeerd kunnen worden. Verdere bijzonderheden vindt u in de beschrijving van **KEY** in hoofdstuk 8.

Heeft u een lang programma, dan voert u in:

KEY 138,CHR$(13)+"CLS:LIST"

Hiermee kunt u het gebied aangeven, dat u wilt LISTen. Tweemaal drukken van de toets, LIST het hele programma.

Wanneer u veel met kleuren experimenteren wilt, kan het gebeuren dat tekst en achtergrond dezelfde kleur hebben en niets meer leesbaar is. Heeft u echter de volgende functie gedefinieerd:

KEY 139,CHR$(13)+"mode 2:ink 1,0:ink 0,9"+CHR$(13)

hierbij hoeft u alleen op de kleinere van de twee [ENTER] toetsen (in het numerieke toetsenveld) te drukken, en u keert naar twee verschillende kleuren terug, zonder dat uw programma verloren gaat.

Deze functietoetsen worden gewist bij het RESETten van de machine. Wanneer u dus uw functietoetsen gedefinieerd heeft, kunt u ze het beste op een cassette vastleggen, zodat u ze altijd weer kunt gebruiken.

1.2.7 Programma's EDITten (inleiding)

U zult zeker fouten maken bij het intikken van programma's. Met de CPC464 is het zeer eenvoudig type-fouten te corrigeren, zonder daarbij per ongeluk regels te overschrijven, die men helemaal niet wil veranderen.

De pijl-toetsen, welke de cursor sturen, helpen u daar te komen, waar de verandering nodig is.

U wilt bijv. in de volgende regel:

10 FOR N = 332 To 255

iets veranderen. Dan heeft u de volgende mogelijkheden:

1. U drukt op [ENTER] en typt de hele regel opnieuw in. De oude regel met hetzelfde regelnummer wordt door de nieuwe, in het geheugen overschreven.

2. Met de toets [←]brengt u de cursor op het te wijzigen teken.

```
10 FOR N = 332 TO 255
```

Het teken,waarop zich de cursor bevindt, is inverse uitgebeeld d.w.z. tekenkleur en achtergrondkleur zijn verwisseld.

Wanneer u nu [CLR] (afkorting voor CLeaR) drukt, wordt het teken gewist en de rest van de regel wordt naar voren geschoven.

```
10 FOR N = 32 TO 255
```

Druk nu [ENTER] en de gecorrigeerde regel is de nieuwe programmaregel. Het speelt geen rol waar de cursor zich daarbij bevindt. De hele regel wordt overgenomen in het geheugen.

3. U kunt de cursor direkt achter het te wissen teken plaatsen:

```
10 FOR N = 332 TO 255
```

Wanneer u nu [DEL] (Afk.voor DELete) drukt wordt het teken links van de CURSOR gewist en de rest van de regel naar links opgeschoven.

Druk [ENTER] en de gewijzigde regel wordt in het geheugen gezet.

```
10 FOR N = 32 TO 255
```

1.2.8. Correcties achteraf

De bovenstaande methoden zijn alleen geschikt wanneer u de fout opmerkt, voordat u de regel met [ENTER] heeft afgesloten. Dit is echter niet altijd het geval,veel fouten merkt u pas op wanneer het programma afloopt en de computer een foutmelding geeft (zie aanhangsel VIII).

Een aantal fouten heeft tot gevolg, dat de foutieve regel op het scherm wordt geprint, waarbij de cursor op het le teken (geheel links) staat. U kunt dan met de boven aangeven methoden de fouten verbeteren, alsof u de regel nog niet met [ENTER] had vastgelegd. Wordt de foutieve regel niet geprint, dan moet u met LIST het programma LISTen, de fout zoeken en met een van de beschreven methodes opheffen.

1.2.9 De Copy Cursor

LIST eerst het programma. Wij gaan ervan uit dat u nog met het korte voorbeeldprogramma werkt, dat in zijn geheel op het scherm past.

```
5 CLS
10 FOR N = 32 TO 255
20 PRINT CHRS(N):
30 NEXT N
```

De fout is in regel 20, waar een S in plaats van een $ staat:($ vertelt de computer dat het volgende teken als tekst en niet als numerieke data moet worden behandeld).

U kunt nu regel 20 opnieuw invoeren of de Screen-Editor(Beeldscherm-bewerking)als volgt gebruiken:

Houdt een van de [SHIFT] toetsen ingedrukt en druk dan de overeenkomstige cursor toets naar beneden [↓] of naar boven [↑], afhankelijk van waar u zijn moet.

De COPY Cursor beweegt zich daardoor weg van de hoofd-cursor (beide hebben dezelfde vorm) en wordt in de regel gebracht die u wilt veranderen. Zet de COPY Cursor op het le teken van de regel.

```
5 Cls
10 FOR N = 32 TO 255
20 PRINT CHRS(N);
30 NEXT N
Ready
■
```

Wanneer u zonder het drukken van de [SHIFT] toets de cursor in de te veranderen regel brengt, heeft dat geen nut, omdat de computer alleen die tekens accepteert die na de hoofdcursor worden ingevoerd.

Zou u dat toch eens doen en op deze wijze de regel overschrijven, kunt u dit eenvoudig corrigeren door de [ESC] toets te drukken <u>VOORDAT U</u> een [ENTER] of functie-toets, die een CHR$(13) inhoudt, indrukt.

Wanneer u toevallig de instructie NEW intypt en met [ENTER] invoert, dan is uw hele programma weg: voorzichtig dus.

Wanneer u een regel verlaten wilt zonder [ENTER] gedrukt te hebben, reageert de computer met een pieptoon. Druk op [ENTER] en het probleem is opgeheven. Mocht u echter een regelnummer ingetikt hebben, dan is deze regel overschreven en moet hij opnieuw worden ingetikt.

Uitgaande van het eerste teken van de regel die u veranderen wilt, gaat u met de toets [COPY] tot aan de positie die u wilt veranderen. Door de herhalingsfunctie (ingedrukt houden van de toets) kan de cursor zeer snel bewogen worden.

```
5 CLS
10 FOR N = 32 TO 255
20 PRINT CHRS(N);
30 NEXT N
Ready
20 PRINT CHR■
```

Laat nu [COPY] los en tik $, deze komt op de plaats waar de hoofdcursor zich bevindt. Deze beweegt zich daardoor een plaats naar rechts.

```
20 PRINT CHR$■
```

Breng nu de COPY Cursor achter de foutieve S, door weer de [SHIFT] toets ingedrukt te houden en eenmaal op de cursor-rechts-toets [→] te drukken. De COPY-Cursor blijft nu op het (staan. Laat de [SHIFT] toets los en druk de [COPY] toets in totdat u over het einde van de regel gekomen bent. Druk nu [ENTER] en de gecorrigeerde regel onder op het beeldscherm vervangt de foutieve regel uit de LISTing.

U kunt met [COPY] de hele regel kopieren en dan gewoon de gekopierde regel verbeteren met de pijltoetsen, [CLR] en [DEL] (zonder [SHIFT]). Het tegelijkertijd indrukken van de toetsen [CTRL] en [←] of [→] brengt de cursor in een keer aan het begin of einde van de regel.

Oefen het, het gaat heel eenvoudig.
Dan is er nog de volgende methode. Tik in:

```
EDIT 20
```

De computer antwoord met:

```
20 PRINT CHRS(N);
```

Gebruik dan de methode zoals in 1.2.7 beschreven is. Wanneer u moeilijkheden heeft, druk dan[ESC], LIST en u heeft het programma terug. De regel waarin [ESC] werd gedrukt, is onveranderd.

Na afsluiting van de correcties typt u LIST en u ziet de nieuwe programma-versie. Wanneer er nog fouten in zijn probeer het nog eens. We hebben nu de basis gelegd voor het werken met de Schneider CPC464

Toets het volgende in, om ook de andere twee modes te leren kennen:

```
MODE 0
RUN
```

Merk op dat het beeldscherm wordt gewist en daarna het programma per regel 20 tekens PRINT.

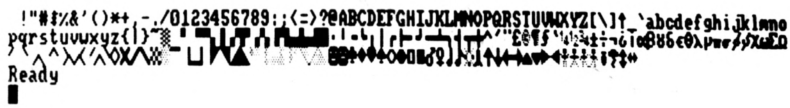


Om tot de voorgaande modus terug te keren, tikt u in:

```
MODE 1
RUN
```

En we zijn terug waar we begonnen. Nu bekijken we de 80 koloms modus, tik:

```
MODE 2
RUN
```



U heeft nu een begin gemaakt. Uw eerste nieuwsgierigheid moet bevredigd zijn. Ervaren gebruikers zullen nu overleggen, welk programma zij als eerte schrijven willen op deze computer. Beginners kunnen het beste verder lezen en zich de specifieke BASIC-instructies eigen maken.

2 DATA RECORDER

Laden en bedienen van de recorder.

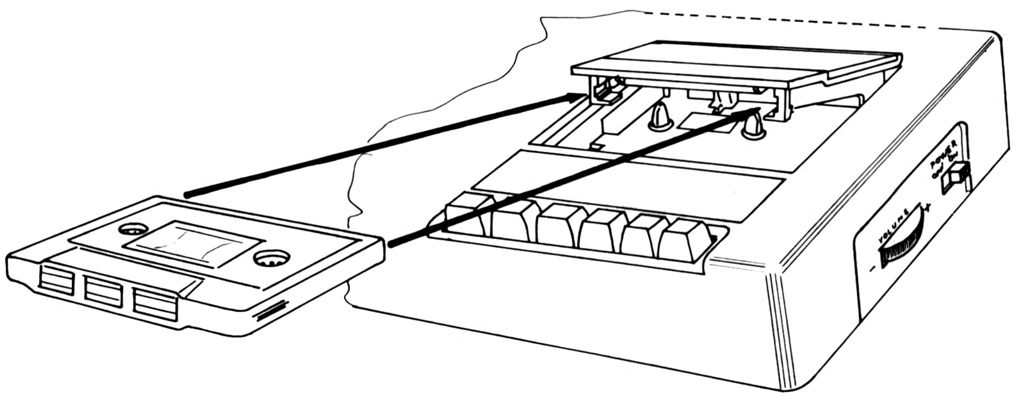
Onderwerpen in dit hoofdstuk:

*** OVEREENKOMSTEN EN VERSCHILLEN TUSSEN DATA- EN AUDIO CASSETTES**
*** LADEN EN RUNNEN VAN DE CASSETTE ; HET DEMO BANDJE**
*** DE KEUZE MOGELIJKHEID VAN DE OPNEEM SNELHEID**
*** OPSLAAN VAN PROGRAMMA'S OP CASSETTE**
*** LEESFOUTEN**

De CPC464 kan alleen data opslaan als hij ingeschakeld en aangesloten op het lichtnet is. In de vaktaal wordt dit tijdelijke opslag genoemd. Wanneer u data en programma´s langer wilt opslaan, moet u dit op cassette of via een aan te koppelen Disk-Drive doen.

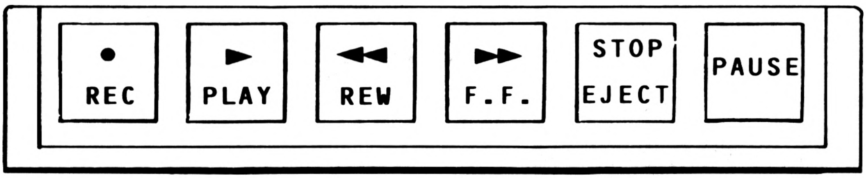
2.1 Bediening van de datarecorder

Rechts van het toetsenbord vindt u de datarecorder (Afb.2.1).
De mechanische opbouw is ongeveer als bij een cassetterecorder, de electronica is echter speciaal op de behoeften voor data-opslag ontworpen.



Afb.2.1: zo wordt de cassette goed ingelegd.

De toetsen en hun functies zijn dezelfde als bij een normale cassette recorder.



Afb.2.2 De toetsen van de CPC464 datarecorder.

Opmerking: deze toetsen moeten harder ingedrukt worden dan die van het toetsenbord.

[REC]= Opname van data of programma's. Bij de opname moeten de opname beveiligingslipjes, achter in de cassette, aanwezig zijn of moeten deze zijn afgeplakt, en het cassettedeksel van de recorder gesloten zijn. Hou de [REC] toets ingedrukt en druk dan op [PLAY]. Nu kan door de opdracht SAVE, via een programma of direkt, op de cassette worden opgenomen.

[PLAY]= afspelen: het laden of lezen van programma's of data van de cassette, door een direkte opdracht of via het programma. Zowel [REC] als [PLAY] worden bij het einde van de band automatisch afgeschakeld.

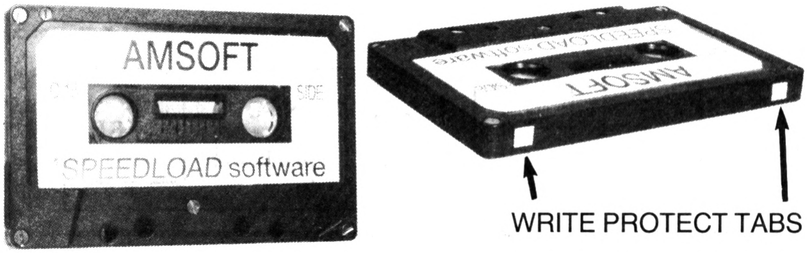
[REW]= terugspoelen van de cassette van rechts naar links. Geen automatische uitschakeling aan het eind van de band. Laat het apparaat in deze functie niet zonder toezicht spoelen, de motor kan schade lijden.

[F.F.]= snel vooruitspoelen naar rechts. Zoals bij [REW] geen uitschakelen van de band aan het einde.

[STOP/EJECT]= stopt alle cassette handelingen en brengt alle toetsen in rusttoestand. Bij tweemaal drukken opent het cassettedeksel.

[PAUZE]= mechanische pauze, te gebruiken bij de [REC] en [PLAY] functie. Gebruik deze toets nooit bij het opnemen of inlezen van data anders komen er fouten. Alle pauzes bij het inlezen of opnemen worden door interne opdrachten van de software gestuurd, de mechanische pauze functie blijft daarbij ongebruikt.

2.2 OPNAMEBEVEILIGING (write protect)



Om het onopzettelijk wissen van cassettes te verhinderen, hebben alle cassettes een mechanische opname-beveiliging. Wanneer u de lipjes met een mes verwijdert, kunt u met deze cassette de [REC] toets niet meer indrukken en daarom is de cassette niet meer overschrijfbaar.

Elke cassette heeft een eigen opnamebeveiliging. Wanneer u op een beveiligde cassette weer opnieuw wilt opnemen, plakt u over de openingen eenvoudig een stukje plakband.

2.3 Laden van de cassette

Afb.2.1 laat zien hoe een cassette goed in de data recorder ingelegd wordt. Spoel met de [REW] toets het bandje helemaal terug. Daarna drukt u op [STOP/EJECT]. Mocht het bandje toevallig uit de opening aan de voorkant van de cassette hangen, spoel dit deel dan met de hand terug tot het weer strak zit, anders kan het bandje kapot gaan en is daarmee uw data verloren.

Behandel de cassette voorzichtig. Een data-cassette is gevoeliger dan een geluids cassette.

Wanneer het bandje beschadigd is, maar u toch nog laden kunt, SAVE dan de data uit het geheugen van de CPC 464 op een andere cassette. Gooi de beschadigde cassette weg zodat u deze niet per ongeluk weer gebruikt.

2.4 Het laden van de Demonstratie-cassette

Met de CPC464 krijgt u een demonstratie cassette bijgeleverd, die een aantal voorbeelden laat zien van de graphics en geluidsmogelijkheden van de computer en zijn ingebouwde software: de Schneider BASIC en het Machine Operating System (MOS).

De software die de recorder de instructies tot lezen en schrijven geeft is een deel van het operating system. LOAD en RUN zijn daarin de meest gebruikte instructies.

Om de bediening van de computer zo eenvoudig mogelijk te maken, zijn in de CPC464 een serie functies ingebouwd, die de instructie invoer zolang vereenvoudigen, totdat u aan het toetsenbord gewend bent. Na het inschakelen van de computer, meldt hij zich met het startbericht en:

Ready

Leg de cassette in de datarecorder zoals in afb.2.1. Zet de teller op 000, door de kleine toets naast het telwerk in te drukken, dan houdt u de [CTRL] en de kleine [ENTER] toets in het numeriek toetsenveld ingedrukt. De volgende mededeling verschijnt:

RUN"
Press PLAY then any key:

(druk op **[PLAY]** en daarna een willekeurige toets)

Ter informatie: Dit is een voorbeeld van het definieren van een functie toets. U heeft daarover in Hoofdstuk 1 kort kunnen lezen: De **KEY** instructie.

U had ook de instructie **RUN"** kunnen intikken en [ENTER] kunnen drukken, maar met het indrukken van 2 toetsen gaat het sneller.

In de meeste gevallen hebben de twee [ENTER] toetsen de zelfde functie. U kunt echter de betekenis van de kleinste van beide toetsen voor speciale gevallen omdefinieren. Op deze vrij definieerbare toetsen komen wij later terug.

De toets [PLAY] kunt u vinden bij de datarecorder: Drukt u hem in tot hij "klikt". De tekst: "Press any key" betekent: "druk een willekeurige toets" en zal beginners wat vreemd in de oren klinken, toch kan dit vaak worden toegepast. Het maakt de bediening van een programma vaak makkelijker.

Beter was het om te zeggen dat u een of andere toets buiten **[SHIFT]**, **[CAPS LOCK]**, **[CTRL]**, **[ESC]** en een toets van de datarecorder in moet drukken. De hier gebruikte tekst geldt voor dit handboek en voor alle door Schneider geleverde programma´s.

De toets die nu indrukt, wordt niet op het beeldscherm weergegeven. Hij start de motor van de data recorder. Zou de band niet starten, probeert u het dan nog eens met een andere toets (het is een goede gewoonte, daarvoor de grote [ENTER] toets te gebruiken). Overtuig u ervan dat de band bij het begin staat, en de [PAUZE] toets niet ingedrukt is.

Ook al drukt u op meer dan een toets, deze worden genegeerd zo gauw het laden van het programma begonnen is.

Let er op dat u geen programma naam heeft ingegeven. Zolang als op de zelfde regel direkt achter

RUN"

niets staat, zal de computer het eerste programma op deze band zoeken en beginnen met opnemen. Als de computer het eerste op deze band goed opgenomen programma gevonden heeft, zal de volgende tekst op uw beeldscherm verschijnen:

Loading WELCOME 1 block 1

Dit bericht vertelt u dat het eerste van een aantal programmablokken, van het programma ´WELCOME 1´ wordt geladen. Ieder programma is in de vorm van samenhangende data blokken ten laste van 2K bytes (dat gelijk is aan 2048 tekens) op de band gezet, die ieder apart gekenmerkt worden. Telkens wordt er een blok ingelezen en zijn nummer op het beeldscherm zichtbaar. Na ieder datablok blijft de data recorder een ogenblik stilstaan, begint dan weer te lopen en kort daarna verschijnt het nummer van het nieuwe blok.

Zou de computer bij het inladen een fout ontdekken dan reageert hij met op het beeldscherm printen van een foutmelding (die vindt u in aanhangsel VIII). In dat geval zit er voor u niets anders op dan het telkens opnieuw te proberen, totdat de computer het programma foutloos inlaadt.

Leest u de aanwijzingen op het beeldscherm, terwijl het programma wordt ingeladen.
De WELCOME-band doet de rest.

2.5 Supersafe en speedload

De CPC464 biedt u twee opnamesnelheden: <u>Supersafe</u>: een overdracht met 1000 baud (bits per seconde) en <u>Speedload</u> met 2000 baud. De tweede snelheid is bij opname en teruglezen dus twee keer zo snel als de eerste. Dit gaat echter ten koste van de zekerheidsmarge. Dit kan bij slechte kwaliteit van goedkope banden en mogelijke fouten op grond van verschillende kopinstellingen bij verschillende recorders fouten veroorzaken.

Voor programma´s die met hetzelfde apparaat opgenomen en ingelezen worden, is de Speedload snelheid zeker goed genoeg, zolang cassettes van goede kwaliteit gebruikt worden. Ook de meeste gekochte prgramma´s kunnen met die snelheid zonder fouten ingelezen worden.

De computer herkent de opnamesnelheid van een gespeelde band en leest met deze snelheid in. Als u met het bevel SAVE een programma wilt bewaren dan moet u de computer laten weten of u de snelle Speedload-snelheid gebruiken wilt, anders wordt er in de langzamere Supersave-snelheid bewaard.

Als u uw programma met Speedload SAVEn wilt, verzekert u er zich dan eerst van dat u in ´direkt mode´ werkt(op het beeldscherm verschijnt het woord **Ready**) en toets dan in:

SPEED WRITE 1

Om naar de (langzame) standaard-snelheid terug te keren, kunt u ten eerste de computer RESETten, in dit geval gaat echter wel alles verloren. U kunt, nadat **Ready** verschenen is,ook intikken:

SPEED WRITE 0

2.6 Het bewaren van programma's en gegevens op cassette

BASIC biedt u verscheidene mogelijkheden, om data en programma´s op cassette te bewaren. Ze worden aan de hand van een voorbeeld hier in het kort getoond.

2.6.1 SAVE "<filenaam>"

De eenvoudigste wijze van opslaan gaat met de instructie SAVE, als de machine zich in de direkte modus bevindt (na het verschijnen van **Ready**, dat na het uitvoeren of LISTen van een programma verschijnt). Neemt u het kleine voorbeeld programma van hoofdstuk 1, dat alle tekens afdrukt.

De <filenaam> kan iedere combinatie van maximaal 16 tekens, inclusief spaties, zijn. Bij langere namen worden de tekens vanaf plaats 16 weggelaten. Als het voorbeeld programma in het geheugen van de computer zit, tik dan in:

SAVE "TEKENS"

De computer antwoordt met de volgende opdracht:

Press REC and PLAY then any key

Herinnert u zich nog de definitie van "Press any key"?. Nu wordt na de uitvoering van deze opdracht het programma met de naam "TEKENS" bewaard.

BELANGRIJK

Als [REC] en [PLAY] niet goed ingedrukt worden en alleen de [PLAY] toets ingedrukt staat, start weliswaar de band, maar het wordt niet beschreven ofschoon het wel zo lijkt.

VOORZICHTIG: als u per ongeluk bij het inladen [REC] <u>en</u> [PLAY] indrukt wordt de band gewist. U kunt als u het snel ondekt, met [ESC] onderbreken, maar het gevaar, dat reeds data gewist is, is groot. Als u zich hiertegen wilt beschermen, moet u er op letten, dat de lipjes aan de achterkant van de cassette verwijderd worden.

Er zijn vier manieren om met de CPC464 gegevens op te slaan. U heeft zojuist de gebruikelijke methode gezien. De drie volgende voor speciale gevallen volgen nu.

2.6.2 SAVE "<filenaam>",A

De handeling verloopt hetzelfde als de vorige keer, echter door de toegevoeging ,A slaat de computer de gegevens niet op in de normale machinetaal maar in de vorm van ASCII tekens.
De wijze voor het opslaan van gegevens wordt normaal gebruikt bij tekstverwerking en andere gebruikersprogramma's. Een gedetaillieerde beschrijving volgt waar deze programma's worden uitgelegd.

2.6.3 SAVE "<filenaam>",P

Met de toevoeging ,P wordt de computer opdracht gegeven de gegevens met een beveiliging op te slaan, zodat het programma niet meer door iedereen eenvoudig gelezen kan worden, als het met RUN" van de cassette in de computer is geladen en de uitvoering met [ESC] is gestopt. Programma's die met een beveiliging zijn geSAVEd kunnen alleen met de RUN" of CHAIN instructies aan het RUNnen gekregen worden. Voordat u deze vorm van opslaan kiest, zou u voor uzelf een onbeveiligde kopie moeten maken.

2.6.4 SAVE"<filenaam>",B,<startadres>,<lengte>[,<beginadres>]

Hiermee kunt u het programma, zoals het in RAM opgeslagen is, in binaire vorm op cassette opnemen. Daartoe heeft de computer de volgende aanwijzing nodig: Op welke plaats in het geheugen bevinden zich de op cassette op te nemen gegevens; hoe lang is deze verzameling van gegevens en eventueel wat het beginadres is, van het programma.

Met deze methode van binaire opslag kunnen gegevens van het beeldscherm direkt op cassette opgeslagen worden. Het nut hiervan kan onder andere zijn; het vertonen van een titelbeeld met als doel het doorbreken van het monotone van een langdurig inlaad-proces.

2.6.5 Files zonder naam en CAT (catalogus)

Als u een file, zonder het een naam te geven, met SAVE "" opneemt, dan zal BASIC het als een naamloze file opslaan. Een cassette kan zo veel files met dezelfde naam of zonder naam opslaan, als er ruimte op de cassette is.

Daarentegen moeten bij het diskettesysteem wel steeds verschillende namen gebruikt worden.
Dat u bij het opslaan zonder naam, snel het overzicht over de cassetteinhoud zult verliezen is duidelijk. Daarom moet u altijd zelfverklarende namen en bij veranderingen toevoegingen, die vertellen welke versie wordt opgeslagen, gebruiken.

U kunt de inhoud van een cassette **CAT**alogiseren, indien u de instructie **CAT** intoetst en de aanwijzingen volgt:

Press PLAY then any key:

BASIC drukt nu de naam van alle op de cassette opgeslagen files af. De filenaam wordt in hoofdletters weergegeven, gevolgd door het nummer van de aanwezige blokken van de file en dan een enkele letter, die de aard van de file aangeeft en het woord **OK** als de data leesbaar en daarmee laadbaar is. De tekens voor de aard van files zijn:

$ standaard BASIC programma
% beschermd BASIC programma
*** ASCII data**
& DATA in binaire vorm

De uitvoering van de CAT instructie heeft geen invloed op het programma dat ondertussen in het geheugen zit.

2.7 Leesfouten

Als u het bericht ontvangt dat een **Read error**(een leesfout) tijdens het laden is voorgekomen, loopt de band weliswaar door en leest de computer de blokken, die hij na degene met de leesfout vindt. Hij zal ze echter niet laden, tenzij hij ze als blok 1 kan indentificeren van het programma dat hij al eerder zonder succes heeft proberen te laden.

Dat wil zeggen dat u na een leesfout met [STOP/EJECT] de band stoppen kunt, met [REW] terug spoelen en op [PLAY] drukken om opnieuw te beginnen. De computer zal nu weer proberen het programma te laden en met een beetje geluk zal dit ook gaan.

Leesfouten kunnen het gevolg zijn van een hele reeks oorzaken. De meest voorkomende zijn mechanische beschadiging van de band. Ook het uitschakelen van de computer tijdens [PLAY] of [REC], kan tot lees fouten leiden.
Er kan dan een spanningspiek via de opname/weergavekop op de band gezet worden als [PLAY] of [REC] en [PLAY] de kop nog tegen de band aangedrukt houden. Het zelfde geldt voor het inschakelen. Ook kan door te lang vasthouden van de band tussen Capstan-aandrijving en aandrukrol de band mmechanisch beschadigd worden.

Verder kunnen leesfouten ontstaan, door het gebruiken van de [PAUZE] toets tijdens SAVEn of LOADen. Tenslotte kan de cassette op een andere CPC464 met niet goed ingestelde opname/weergave kop opgenomen zijn.

Leesfouten kunnen soms geheel toevallig ontstaan. Oorspronkelijk werden cassettes niet voor dataopslag, maar voor de geluidweergave ontwikkeld. Dat maakt het opslaan lastiger t.o.v. de wezenlijk duurdere profesionele systemen.

Niettemin is de cassette een prima opslagmedium voor een zeer gunstige prijs. De grootte van de magneet-deeltjes op de band stelt echter samen met de bandsnelheid en de snelheid van de data overdracht tussen band en computer fysische beperkingen. Elke poging om met snelheden boven de tolerantiegrens van het systeem te werken, zal tot onbetrouwbare resultaten leiden, met name bij massavermenigvuldiging, die bij de productie van goedkope software gebruikt wordt.

Opmerking: cassettes met programma's van een ander computersysteem kunt u met de CPC464 niet lezen. Ze kunnen er precies hetzelfde uitzien, bij het afspelen met een normale cassette recorder gelijksoortige geluiden produceren, maar ze kunnen niet geladen en geRUNd worden. Zou het u toch lukken, verzoeken wij u om ons een precieze beschrijving van het computertype en het programma toe te zenden.

2.8 CASSETTE'S, welke wordt gekozen

Ofschoon u natuurlijk elk cassette-type (tot en met C90) gebruiken kunt, geven wij de voorkeur aan C12 (6 minuten per zijde) tot ten hoogste C30 cassettes. Hoe langer (C60/C90) een cassette is des te langer duurt het natuurlijk, voordat een programma (vooropgesteld dat u de filenaam weet), gevonden en geladen wordt. Een andere mogelijkheid is, steeds de stand van de teller in de filenaam op te nemen. Bij het overschrijven van een programma aan het einde van een lange band, moet u het beginpunt op de band zorgvuldig bepalen om te zorgen, dat u niet per ongeluk delen van een ander programma overschrijft.

Al met al is het het beste zo veel mogelijk bandjes met zo weinig mogelijk programma's te laden. C12 cassette's zijn relatief goedkoper en daarom is ook de verleiding gering, een cassette die beschadigd, maar nog ten dele in orde is, te blijven gebruiken.

En denkt u er alstublieft aan, dat op alle commerciele software COPYRIGHT staat. Het is verboden om gekochte software op een of andere manier te reproduceren, ook niet voor een vriend, hetzij dat de verkoopvoorwaarden het toestaan. (sommige programma's vragen er zelfs om, voor uzelf een copy te maken). Software diefstal is strafbaar en de wetten woorden voortdurend herzien om dergelijke praktijken tegen te gaan. Worden programma's erg vaak gekopieerd, dan is het voor de originele makers niet meer lonend nieuwe programma's te ontwikkelen. Het gevolg kan zijn, dat het programma aanbod voor uw computer erg mager wordt en daar is niemand mee gediend.

3. BASIC INLEIDING

Een korte inleiding voor programma's, die in Schneider BASIC geschreven zijn

Onderwerpen in dit hoofdstuk

* **Syntax-regels en hun beschrijving**
* **PRINT-opdrachten, gegevensverwerking en formattering op het beeldscherm**
* **ZONE's**

3.1 BASIC beginselen

Het fundamentele verband tussen CPC464 BASIC en de interne verwerking is in aanhangsel II beschreven. Als U nog nooit geprogrammeerd heeft, willen wij u daar bij helpen - maar soms worden bepaalde begrippen als bekend verondersteld, die de beginner
moeilijkheden kunnen bezorgen. Zou dat op u betrekking hebben, kunt u het beste een boek lezen, dat u enige basiskennis over microcomputers bijbrengt.

U moet aan de hand van de eenvoudige voorbeelden het volgende hoofdstuk doornemen. Als u niet meteen alles begrijpt, is dat niet erg, maar hoe meer u leert, des te gemakkelijker begrijpt u de daarop volgende onderwerpen.

BASIC is de taal waarmee u in uw CPC464 werkt. Zodra u inschakelt staat de BASIC voor u gereed. Dat ziet u aan de mededeling:

Ready

BASIC is de gemakkelijkst te leren programmeertaal. Het bestaat uit duidelijk gedefinieerde begrippen met eenvoudige regels. Zodra u deze begrijpt, heeft de BASIC voor u geen geheimen meer.

Schneider BASIC kan de in hoofdstuk 8 beschreven instrucies uitvoeren. Iedere instructie is door een of meerdere BASIC woorden gedefinieerd en kan verscheidene parameters hebben. Die kunnen soms vrij worden bepaald. In zijn algemeenheid kan een parameter bestaan uit, constanten, variabelen of functies. Combinaties van letters en cijfers worden strings genoemd. Verschillende vormen van getalweergave zijn mogelijk: decimaal, hexadecimaal en binair.

Op cassettes opgeslagen gegevens worden altijd sequentieel, dat wil zeggen, de een na de ander: van voor naar achter, ingelezen, in tegenstelling tot willekeurig toegankelijke bestanden, de zgn. Random access Files.

3.2 De structuur van een BASIC programma

Opdrachten worden in BASIC, regel voor regel ingevoerd. Een regel kan meerdere opdrachten bevatten, die door dubbele punten gescheiden worden. Maximaal zijn 255 tekens per regel mogelijk. Een teken is hier een letter, cijfer of spatie. In de direkte modus worden opdrachten direkt van het toetsenbord ingevoerd en mogen zij geen regelnummers bevatten. In programma-modus worden de opdrachten uit het geheugen in de volgorde van de regelnummers uitgevoerd.

Schneider BASIC maakt het de gebruiker mogelijk, in direct modus programmaregels toe te voegen of weg te laten of bestaande regels te verbeteren. Voordat een programma geLIST of geRUNd wordt, sorteert de computer de regels overeenkomstig met de regelnummers. Het maakt niet uit in welke volgorde ingetoetst werd.

3.3 Regels intikken

BASIC accepteert regels tot een lengte van 255 tekens gevolgd door [ENTER]. Tijdens het intoetsen is het mogelijk, de regel die men aan het intikken is te veranderen en met de COPY Cursor-functie, tekens uit andere regels op het beeldscherm in te voegen; zie Hoofdstuk 1.2.7.

Alle BASIC woorden moeten met een scheidingsteken beeindigd worden. Dat kan een spatie, een +, -, of een ander geldig scheidings- teken zijn. Dat is nodig, omdat BASIC woorden, gereserveerde woorden, delen van variabelen kunnen zijn - aan een gereserveerd woord moeten andere tekens worden toegevoegd, om als variabelenaam geaccepteerd te worden.

BASIC woorden kunnen in kleine of, met de [SHIFT] toets ingedrukt, in HOOFDLETTERS geschreven worden.

De PRINT instructie kan door een vraagteken vervangen worden. In dit geval is geen scheidingsteken nodig. Wiskundige tekens (**+-*/MOD**) zijn ook BASIC woorden. Het volgende voorbeeld is geldig, maar niet aan te raden, omdat het tot slechte gewoonten leidt bij het programmeren. Spaties zijn hier beter:

```
for n= 1 to 50:?n:next
```

Ook een enkelvoudig aanhalingsteken ' (**[SHIFT]7**) kan als kommentaar, de **REM** instructie vervangen.

Met aanvullende spaties wordt geen rekening gehouden. Ze dienen alleen voor betere leesbaarheid van een programma.

3.4 Terminologie

Om BASIC instructies en opdrachten te beschrijven, is een formele maar eenvoudige terminologie nodig. Iedere instructie moet exact zo worden ingetoetst. Variabelen of toevoegingen, die al of niet kunnen worden gebruikt, worden door "plaatsaanduiders" (verschillende soorten haakjes) aangeduid. Een gedetailleerde beschrijving volgt in het volgende deel.

De toevoegingen worden door verschillende namen gekenmerkt en in groter-dan en kleiner-dan tekens <> weergegeven. Op verschillende plaatsen is bijvoorbeeld een getal vereist. Dit wordt als volgt weergegeven:

<getal>

Alles wat niet tussen deze tekens staat, moet precies zo ingevoerd worden als voorgeschreven is. De instructie STOP heeft bijvoorbeeld de volgende vorm:

STOP

Als in de definitie een toevoeging, waarvan de keuze vrij is, staat, wordt dit door rechte haken [] aangegeven. Als voorbeeld hier een al of niet te gebruiken getal:

[<Getal>]

Als zo een toevoeging, waarvan de keuze vrij is, meerdere malen (dat wil zeggen zo vaak als men wil of helemaal niet) voorkomen kan, wordt dit door een sterretje * na de rechte sluithaak] aangegeven. Een getallenreeks, die uit minstens een getal bestaan moet, wordt bijvoorbeeld zo weergegeven:

<Getal>[<Getal>]*

Zo 'n uitdrukking kan er als volgt uitzien:

34
of **344**
of **345678** etc.

In veel gevallen worden opties in een reeks steeds door komma's gescheiden. De gebruikte korte vorm wordt het best aan de hand van een voorbeeld duidelijk:

<Reeks van:<Uitdrukking> betekent: <uitdrukking>[,<Uitdrukking >]*
of
<Reeks van:[#]<Getal> betekent: [#]<Getal>[,[#]<Getal>]*

Een voorbeeld daarvan is:

3,4
of **3,4,4**
of **3,4,5,6,7,8** etc.

Zo'n reeks kan ook een enkel gegeven zijn. Als de reeks meer dan een gegeven inhoudt, moet ieder toegevoegd gegeven met een ervoor gezette komma toegevoegd worden, want dit teken legt de grens tussen de door de computer apart te verwerken gegevens vast.

Getallen kunnen in verschillende vormen weergegeven worden:

a. <getallen in de gewone vorm>
   d.w.z. getallen zonder exponent

b. <getallen in de wetenschappelijke notatie>
   d.w.z. in de vorm: 2E4 (2 maal 10-tot-de-macht-4)
   de exponent kan positief zijn of negatief.

c. <Getallen in hexadecimale of binaire vorm> (zie aanhangsel II)
Decimale vorm (normale vorm) 100
Hexadecimale vorm &64 of **&H64** (De H is keuzevrij)
Binaire vorm **&X1100100** (De X is verplicht)

3.5 Oefening baart kunst: de PRINT-instructie

Als demonstratie hoe BASIC werkt worden hier enkele voorbeelden getoond.

Een BASIC woord, dat zeer veel mogelijkheden voor de opbouw van BASIC opdrachten toont, is **PRINT**. Een instructie is een BASIC woord, dat direct of in de programma-modus van de computer uitgevoerd wordt. Om tot uitvoeren van een functie te komen is een instructie nodig. Een voorbeeld:

```
PRINT FRE("")
```

Opdat de CPC464 op een vraag beantwoorden kan, moet u hem drie dingen zeggen:

1. De plaats waar het antwoord verschijnen moet: printer, beeldscherm of elders
2. U moet de computer de data geven, waarmee hij moet werken.
3. U moet de computer zeggen, wat hij met de data moet doen.

Met PRINT wordt de computer gevraagd, de resultaten van een instructie naar een bepaald randapparaat (stream) te schrijven, waarbij stream door een getal tussen 0 en 9 wordt aangeduid. In de BASIC beschrijving wordt dit als <STREAM-uitdrukking>(stream expression) gekenmerkt. De stream-uitdrukking legt vast, welk randapparaat moet worden gebruikt:

0...7 zijn tekststreams, die naar zogenaamde tekst-vensters (WINDOWS) voeren. Deze werden van te voren met **WINDOW** gedefinieerd.

8 is de parallelle printer-uitgang. Deze kan alleen gebruikt worden als een printer met Centronics parallel aansluiting correct aangesloten is.

9 Is een uitvoerkanaal naar de cassetterecorder, die vooraf in het programma geopend moet zijn.

De korte vorm van de PRINT instructie (met uitzondering van PRINT USING) is:

```
PRINT [#<randapparaat>][<af te drukken gegevens (print list)>]
```

De rechte haken betekenen, dat deze toevoegingen voor een geldige instructie niet echt noodzakelijk zijn. Als u alleen PRINT invoert, wordt een lege regel op het beeldscherm gezet: probeert u het maar eens. Als u geen bepaald apparaat voor uitvoer aangeeft, kiest de CPC464 uit zichzelf apparaat **#0**, beeldscherm onder de cursor. Probeert u deze instructie, vergeet u daarbij niet, aan het einde van de opdracht [ENTER] in te voeren, zodat de computer weet, dat hij deze opdracht nu moet uitvoeren:

```
PRINT "HALLO"
```

Het resultaat is:

HALLO

U ziet dat de aanhalingstekens niet mee afgedrukt worden. Deze dubbele aanhalingstekens worden bij BASIC gebruikt, om begin en einde van een <print list> te bepalen.

Voert u nu in:

PRINT #0,"HALLO"

en u krijgt hetzelfde resultaat.
Maar als u ingeeft:

PRINT #4,"HALLO"

...dan verschijnt het resultaat geheel links boven op het beeldscherm. Dit is de eerste plaats voor tekstvenster 4, dat zich normaal over het gehele beeldscherm uitstrekt, als u niet voorheen met **WINDOW** iets anders heeft bepaald. De startpositie voor ieder tekstvenster (WINDOW) is links boven en stream #4 was nog niet gebruikt. Als deze instructie nu nogmaals zou worden uitgevoerd, stond het tweede HALLO onder het eerste. Dit geldt voor ieder WINDOW. Daarom werd bij de instructie zonder streamtoevoeging HALLO op de momentele cursor positie geschreven.

Deze functie van Schneider BASIC is bijzonder krachtig, omdat het de opbouw van komplexe WINDOWS met eenvoudige instructies en definities mogelijk maakt.

BASIC drukt alles af, wat binnen de aanhalingstekens staat, zonder op de inhoud te letten. Zelfs gereserveerde woorden zijn daarbij toegestaan:

PRINT "4*4"

Het antwoord is:

4*4

Als u wilt, dat BASIC de rekenopgave uitvoert (4 keer 4), dan moeten de aanhalingstekens weggelaten worden.

PRINT 4*4

Het resultaat is dan

16

Let u erop, dat het getal 1 kolom naar rechts van de linker rand wordt geschreven, daar BASIC bij een negatief resultaat daar het minteken zou zetten.

De PRINT instructie heeft nog veel andere vormen, die de volle formatterings mogelijkheden van de standaardserie modelvormen benutten.

De <print list> in de PRINT instructie heeft betrekking op datgene, wat afgedrukt moet worden. Dat kan een getal, een variabele, een stringexpressie (d.w.z.een voorheen gedefinieerde stringvariabele, b.v. HALLO$), of iets anders wat gedrukt kan worden tussen aanhalings- tekens, zijn.

PRINT USING ordent de getallen in een vast formaat, zodat koloms gewijze afgedrukt wordt en achter de komma alles of een deel weg-gelaten kan worden.

3.6 PRINT USING en ZONE

Als de CPC464 ingeschakeld wordt, verdeelt BASIC het beeldscherm in zones, die ieder 13 kolommen breed zijn (vergelijkbaar met een tabulator). Als de <print list> komma's tussen de verschillende delen heeft staan, wordt telkens het volgende element vanaf de volgende ZONE afgedrukt. Als niet meer zoveel kolommen over zijn, als in de ZONE-instructie aangegeven is, wordt het volgende element op een nieuwe regel geschreven.

Zonder het USING formaat drukt BASIC positieve getallen met een open plaats, negatieve getallen met een min-teken voor het getal. Ieder getal wordt gevolgd door een open plaats. De decimale punt wordt niet afgedrukt, als er geen waarde achter staat.

Schneider BASIC gebruikt de **[TAB]** toets niet als kolommen tabulator, omdat deze functie niet genoeg gestandaardiseerd is en in de verschillende BASIC versies, verschillend gebruikt wordt. Als men de [TAB] toets indrukt, wordt een pijl-rechts-teken [→] gegeven (zoals bij het gelijktijdig drukken van de [CTRL] en I toets). Verder heeft [TAB] geen betekenis in Schneider BASIC.

3.7 PRINT TAB (<geheelgetal expressie>)(<print list>)

De uitwerking van dit commando kan het beste aan de hand van een voorbeeld getoond worden. Voert u dit voorbeeld programma in de computer:

```
5 MODE 2: INK 1,0: INK 0,9
10 FOR N=1 TO 5
20 ZONE 40
30 PRINT TAB(N*4)"HI",N
40 NEXT
```

Dit programma demonstreert zowel ZONE samen met komma , als de functie van **TAB ()**. Laat u het programma nog eens lopen, maar verandert u eerst regel 10 als volgt:

```
10 FOR N=-5 TO 5
```

De TAB instructie zorgt er voor, dat PRINT na zoveel spaties begint, zoals in <geheelgetal expressie> (**integer expression**) gedefinieerd wordt. ZONE kan van 1 tot en met 255 gedefinieerd worden. Zie beschrijving in Hoofdstuk 8.

Het formaat PRINT USING wordt gebruikt, om gewone getallen in een beter leesbare vorm te drukken. Het is een beetje ingewikkeld te gebruiken en u kunt dit het beste met praktische voorbeelden uitproberen, daar zijn vorm:

```
PRINT[#<randapparaat>,][<print list>][<using-clausule>]
      [<scheidings- teken>]
```

.gebruikers onvriendelijk genoemd kan worden, te meer daar de **<using-clausule>** verder onderverdeeld kan worden in

```
USING <string expressie>;[<using list>]
```

waarbij de **<using-list>** verder onderverdeeld wordt in:

```
<expressie>[<scheidings-teken><expressie>]*
```

Probeert u het volgende voorbeeld:

```
PRINT 123.456, USING "###.##";4567.896
```

Het resultaat is dan:

```
123.456   %4567.90
```

Hier worden verschillende zaken gedemonstreerd. Ten eerste: het getal werd zonder USING ongewijzigd afgedrukt. Ten tweede: USING bepaalt, over hoeveel posities het af te drukken gegeven wordt geprint. De af te drukken informatie kan langer zijn dan de toegewezen posities. Als het links van de komma gebeurt, wordt het getal weliswaar gePRINT, maar een %-teken wijst er op, dat het getal te groot is.
Vervolgens zorgt de komma na **123.456** ervoor, dat het volgende getal aan het begin van de volgende print-ZONE geschreven wordt. Zou in plaats van de komma een puntkomma gebruikt worden, dan zou de afdruk na een enkele spatie achter **123.456** worden gezet.
Getallen worden voor de duidelijkheid altijd d.m.v. een spatie gescheiden als ze in dezelfde regel geschreven worden.

Let u er tenslotte op dat het resultaat afgerond is. De overtollige decimale plaatsen worden dus niet zonder meer vergeten.

Probeert u dit:

```
PRINT 123.456, USING "#####.##+";4567.899
```

en u ziet het plusteken aan het einde van het geformatteerde getal. Een minteken wordt bij een negatief getal automatisch afgedrukt.

PRINT USING is een prima mogelijkheid om formulieren en lijsten te drukken. U wordt ook altijd door een **%** er op gewezen, als het getal te groot is.

4. VARIABELEN, OPERATOREN en GEGEVENS

OMGANG MET GEGEVENS IN EEN BASIC PROGRAMMA

Onderwerpen in dit hoofdstuk:

* Type variabelen: REAL,INTEGER en STRING
* Operatoren, logische uitdrukkingen
* Reeksen
* Gegevens

4.1 Gereserveerde woorden

U heeft al gemerkt dat instructies en andere gereserveerde woorden in Schneider BASIC begrensd moeten worden door scheidingstekens, het zij door spaties, leestekens dan wel numerieke operatoren enz.. Programma's zijn zo makkelijk te lezen en uit te testen, omdat deze woorden zelfs als ze met kleine letters worden ingevoert, bij het LISTen in hoofdletters worden afgedrukt. Gebeurt dit niet dan wordt hiermee aangegeven, dat een instructie fout werd ingevoerd en zal het programma niet goed lopen. Bij Schneider BASIC kan men BASIC woorden als een deel van variabelenamen gebruiken. Een variabele is een naam die u aan een bepaald geheugendeel toekent. Dat kan een enkele letter zijn (variabelen moeten altijd met een letter beginnen), maar in een lang programma is het beter leesbaar, als u zelfverklarende namen voor uw variabelen gebruikt:

```
ANTWOORD=4*4: print ANTWOORD
```

In Schneider BASIC kunnen variabelen tot en met 40 tekens lang zijn waarbij het eerste teken een letter moet zijn. Spaties zijn verboden. BASIC leest anders de variabele maar tot aan de spatie en meldt dat er een fout is:

```
Syntax error
```

Dit betekent dat een ongeldige rij tekens ingevoerd is (zie aanhangsel VIII). Als u een uitdrukking bestaande uit meerdere woorden wil gebruiken, zet u een punt in plaats van een spatie tussen de diverse elementen. Alle gebruikelijke vormen van toegewezen variabelen zijn mogelijk. Maar denkt u eraan reeksen met DIM(ension) te definieren.

4.2 Beknopte schrijfwijze

Omdat het lastig is, steeds opnieuw PRINT te moeten intikken, kunt u daarvoor een ? gebruiken. BASIC begrijpt dat daarmee PRINT bedoeld wordt (zolang u het niet tussen aanhaling tekens " " zet).
Let er wel op dat na het ? geen spatie hoeft te volgen, zoals dit wel het geval was bij PRINT. Als u de volgende regel als programma schrijft:

```
10?4*4
run
```

Is het antwoord hetzelfde. Als u nu het programma LIST...

```
list
10 print 4*4
```

..ziet u dat BASIC, volgens de regels, daaruit een PRINT instructie heeft gemaakt. In PRINT instructies, waarbij aanhalingstekens voorkomen...

```
10? "HALLO"
```

..kunt u ook lui zijn en het laatste aanhalingsteken in een regel weglaten, zoals bij de functie RUN", die door het indrukken van [CTRL] en kleine [ENTER] geproduceerd wordt. Het zelfde functioneert het in programmaregels, maar is niet aan te raden wegens de moeilijkheden, die bij toevoegingen aan de regel kunnen onstaan, als u vergeet met aanhalingstekens af te sluiten.

4.3 Meerdere statements in een regel en samengestelde berekeningen

U kunt meerdere instructies in een BASIC regel schrijven. Een BASIC regel kan tot en met 255 tekens lang zijn en heeft niets met de regel begrenzing van het beeldscherm te maken. De instructies moet u van elkaar scheiden door een dubbele punt **:**.

```
?2*8/5+5-4*777E9/3
```

geeft als antwoord:

```
-1.036E+12
```

Belangrijk is het de volgorde te begrijpen waarin de afzonderlijke wiskundige uitdrukkingen (**+-*/MOD** enz) achtereenvolgens door BASIC afgewerkt worden. U zou anders nooit het goede rekenresultaat verkrijgen. Dit is de volgorde:

^ machtsverheffen
* vermenigvuldigen
/ delen
\ geheelgetal deling: alleen het geheel getal wordt als uitkomst gegeven
\+ optellen
\- aftrekken

Alles wat tussen haakjes ( ) staat wordt het eerste behandeld. Als binnen de haakjes samengestelde berekeningen staan, dan gelden daarvoor bovenstaande regels inclusief de haakjesregel, als binnen de haakjes weer haakjes zouden voorkomen. U moet ze steeds per paar gebruiken, anders krijgt u een: **Syntax error**

4.4 Verdere stappen

U heeft nu al heel wat geleerd sinds dat u in hoofdstuk 3.5 met de PRINT instructie heeft kennis gemaakt. De grondregels van Schneider BASIC, die u nu moet kennen, maken het u mogelijk, een verdere stap in de richting van echte programmering te doen. Voor de opgaven die we tot nu toe gemaakt hebben, hadden we ook een eenvoudig rekenmachientje kunnen gebruiken. U wordt nu met de nodige BASIC instructies vertrouwd gemaakt (in hoofdstuk 8 vindt u de precieze beschrijving en functie in alfabetische volgorde).

Vele BASIC woorden vertellen meteen wat ze betekenen, tenminste in de Engelse taal. GOTO 50 betekent: spring naar regel 50 en ga daar door met de bewerking. END betekent: einde BASIC bewerking en keer in de direkt mode terug en print READY.

In de direkte mode kunt u een reeks van opdrachten invoeren die door een dubbele punt : gescheiden zijn. Maar zodra u echter [ENTER] heeft gedrukt, worden deze instructies uitgevoerd en die BASIC regel staat dan niet meer te uwer beschikking. Met behulp van de COPY cursor kunt u echter de instructies nog eens laten uitvoeren, vooropgesteld dat de regel nog op het beeldscherm staat.

4.5 Voorwaardelijke en logische statements

BASIC benut de mogelijkheden van de computer: steeds terugkerende opgaven met hoge snelheid uit te voeren en zonder dat het hem gaat vervelen. Een aantal instructies zijn voorradig, om deze zogenaamde lus-techniek (looping) mogelijk te maken. Begin, controle en beeindiging van een lus zijn derhalve zeer eenvoudig.

De laatste van deze elementen zijn de vergelijkings voorwaarden. Hoe verschillende gegevens zich tot elkaar verhouden, wordt door het vergelijken vastgelegd. U kunt met variabelen of met vaste gegevens vergelijken.
De vergelijkings operatoren zijn:

```
<  kleiner dan
<= kleiner dan of gelijk aan
=  gelijk aan
>  groter dan
>= groter dan of gelijk aan
<> ongelijk aan
```

Hier komt een klein programma dat het gebruik van deze operatoren aan de hand van een voorbeeld, dat ons allen na aan het hart ligt, laat zien.
Als u dit beeld nog niet op u beeldscherm heeft:

```
Schneider 64K Microcomputer (v1)
© 1984 Amstrad Consumer Electronics plc
          and Locomotive Software Ltd
BASIC 1.0

Ready
```

Drukt u nu op [CTRL], [SHIFT] en [ESC]. U weet toch, dat daarmee de computer geRESET wordt. Nu ziet u het bovenstaande beeld. Voert u nu het volgende programma in (hoofdstuk 1.2.7 behandelt het intikken van programma regels):

```
10 INPUT "hoe hoog is uw inkomen";INKOMEN
20 IF INKOMEN < 2000 THEN GOTO 30 ELSE 40
30 PRINT "vraag om opslag" :END
40 PRINT "vraag om een grotere auto"
```

Laat u dit programma lopen, door RUN te tikken, gevolgd door [ENTER]. De computer stelt u nu de volgende vraag:

```
hoe hoog is uw inkomen?
```

(let er op dat bij vragen de computer automatisch een vraagteken zet als het van u invoer verwacht). Geeft u nu alleen maar cijfers in: geen letters, valutatekens of komma´s en eindig zoals altijd met [ENTER].

Voegt u nu de volgende regel toe, zodra Ready verschijnt.

```
5 CLS
```

Voert u met RUN het programma nogmaals uit, nu wordt het scherm gewist. Was uw eerste antwoord kleiner dan 2000, dan neemt u deze keer een groter getal dan 2000 en u ziet het verschil. Voegt u nu na READY regel 50 toe.

```
5 CLS
10 INPUT "hoe hoog is uw inkomen"; INKOMEN
20 IF INKOMEN < 2000 THEN GOTO 30 ELSE 40
30 PRINT "vraag om opslag": END
40 PRINT "vraag om een grotere auto"
50 IF INKOMEN >4000 THEN PRINT" en zoek een goeie accountant"
RUN
```

De END instructie in regel 30 stopt het programma. In regel 40 is geen END gegeven, daarom gaat het programma daar door en controleert of de variabele INKOMEN groter dan 4000 is. Is dit het geval dan geeft het programma nog een goede raad.

Gaat u verder en voer nog regel 60 in:

```
5 CLS
10 INPUT "Hoe hoog is uw inkomen"; INKOMEN
20 IF INKOMEN < 2000 THEN GOTO 30 ELSE 40
30 PRINT "Vraag om opslag": END
40 PRINT "Vraag om een grotere auto"
50 IF INKOMEN >4000 THEN PRINT " en zoek een goeie accountant"
60 IF INKOMEN >3000 THEN PRINT "...en leent u mij 50 gulden"
run
```

U ziet dat, de tekens (> en <) hier worden gebruikt om de grenswaarden te bepalen. U hoeft ook geen spatie er voor of er achter in te voeren. RUNt u het programma met als inkomen 3600, dan ziet u dat regel 50 wordt overgeslagen, omdat de voorwaarde niet opgaat, maar dat regel 60 wordt uitgevoerd, omdat hier de voorwaarde goed is.

Vanaf dit punt kunt u met de tot nu toe verkregen kennis een eigen programma schrijven. Kijkt u goed, hoe dit programma gegroeid is. De meeste programma´s onstaan zo en hier ligt een van de sterkste kanten van BASIC, omdat dit het ontwerpen zo makkelijk maakt.

4.6 Programma's ontwerpen

De wijze waarop in BASIC programma´s kunnen worden ontworpen is waarschijnlijk het grootste voordeel. Degene, die gestructureerd werken ,zullen beweren dat het steeds maar toevoegen van nieuwe gegevens tot onoverzichtelijke programma´s leidt. In werkelijkheid heeft het echter grote voordelen. Speciaal als u begint te programmeren en niet overziet, hoe het programma uiteindelijk moet worden. Door de direkte uitvoering heeft u meteen een controle op het resultaat.

Neemt u het voorbeeldprogramma op en voegt u regel 70 toe. Daarmee wordt een vertraging in het programma ingebouwd, die u enige tijd geeft het bericht op het beeldscherm te kunnen lezen.

```
5 CLS
10 INPUT "Hoe hoog is uw inkomen";INKOMEN
20 IF INKOMEN < 2000 THEN GOTO 30 ELSE 40
30 PRINT "Vraag om opslag":GOTO 70
40 PRINT "Vraag om een grotere auto"
50 IF INKOMEN >4000 THEN PRINT "en zoek een goeie accountant"
60 IF INKOMEN >3000 THEN PRINT "...en leent u mij 50 gulden"
70 FOR n=1 TO 900:NEXT n: GOTO 5
run
```

Let op dat de regels in kleine letters zijn ingetikt. Schneider BASIC maakt onderscheid tussen BASIC-woorden en variabelenamen. Drukt u [ESC] tweemaal in, om het programma af te breken. LIST nu het programma. Alle BASIC-woorden zijn nu in hoofdletters, behalve de variabele n

Regel 70 bevat een vertragingslus, waarin het programma eerst van 1 tot 900 moet tellen, voordat de volgende instructie wordt uitgevoerd. Het programma komt nu door GOTO 5 in een lus zonder eind. U kunt het programma alleen stoppen door op [ESC] te drukken. Na een keer drukken pauzeert het programma. Na twee keer drukken keert BASIC in de direkt mode terug. Het programma is echter nog altijd in het geheugen.

Als u [ESC] drukt terwijl het programma op invoer wacht wordt het meteen onderbroken. Het regelnummer, waarin het programma stopt, wordt in de **BREAK** melding getoond.

Break in 10

Als u het programma gedurende de wachtlus in regel 70 onderbreekt (tweemaal [ESC]) dan komt de melding:

Break in 70

Als u het lopende programma in regel 70 laat pauzeren, dan kunt u het vanaf deze plaats verder laten lopen. Door op een of andere toets te drukken. Als u het programma geheel afgebroken heeft (tweemaal [ESC]) kunt u het toch verder laten gaan door:

CONT

in te voeren. Wat u ook met de [ESC] toets doet, het programma in het geheugen gaat niet verloren. Alleen als u NEW invoert of de computer RESET (**[CTRL]**, **[SHIFT]** en **[ESC]**), dan is het gewist.

Er is dus geen dubbele veiligheid nodig voor degene die per ongeluk de machine RESET. Het wissen uit het programmageheugen gebeurt altijd bewust en definitief. Zet u alles op cassette wat u misschien nog eens gebruiken kunt.

4.7 Meer over variabelen en strings

Het werken met variabelen is de basis van het programmeren. Zouden alleen vaste waarden worden verwerkt, dan zou de computer alleen een electronische schrijfhulp zijn. Is een deel van een wiskundige berekening een variabele, dan is de rest ook variabel. Variabelen hebben 3 eigenschappen

- Naam
- Soort
- Organisatie

Namen voor variabelen werden in 4.1 behandeld. Het type is naar keuze. U kunt een variabele volgens de regels van 3.4 als volgt definieren:

<naam>[<soortkenmerk>]

Soortkenmerken zijn:

% voor hele getallen (integer). Cijfers achter de komma worden genegeerd. Integer variabelen hebben weinig plaats nodig in het geheugen. Als in een programma geen breuken verwerkt worden, kunnen met DEFINT de variabelen als integervariabelen worden gedefinieerd. Het waardebereik van de integer getallen is -32768...+32767.

! definieert een variabele als reeel getal (REAL). Reele getallen bestaan uit cijfers voor en achter de komma. Na het inschakelen van een computer worden alle variabelen als reeel beschouwd. Daarom moet u variabelen alleen dan als reeel definieren, als u eerder DEFINT gebruikt heeft. Reele getallen kunnen iedere waarde in het bereik van 2.9E-39 to 1.7E+38 aannemen.

$ definieert een stringvariabele. De inhoud kan bestaan uit alle mogelijke tekens tussen aanhalingstekens.
Bijvoorbeeld:

```
NAAM$="Dirk Hagen"
```

Voeg nu regel 6 aan het voorbeeldprogramma toe en verander regel 60:

```
5 CLS
6 INPUT "Hoe heet u";NAAM$
10 INPUT "Hoe hoog is uw inkomen";INKOMEN
20 IF INKOMEN < 2000 THEN GOTO 30 ELSE 40
30 PRINT "Vraag om opslag":GOTO 70
40 PRINT "Vraag om een grotere auto"
50 IF INKOMEN >4000 THEN PRINT " en zoek een goeie accountant"
60 IF INKOMEN >3000 THEN PRINT "...en leent u mij 50 gulden "; NAAM$
70 FOR n=1 TO 900:NEXT n: GOTO 5
run
```

Een spatie werd na het woord **gulden** ingevoegd om te vermijden, dat de naam direkt naast **gulden** wordt afgedrukt. De puntkomma **;** na de PRINT en INPUT instructies verhindert, dat na deze instructies een nieuwe regel begonnen wordt.
We kunnen ook met hele getallen (integer) werken. Voert u regel 61 in:

```
5 CLS
6 INPUT "Hoe heet u";NAAM$
10 INPUT "Hoe hoog is uw inkomen";INKOMEN
20 IF INKOMEN < 2000 THEN GOTO 30 ELSE 40
30 PRINT "Vraag om opslag":GOTO 70
40 PRINT "Vraag om een grotere auto"
50 IF INKOMEN >4001 THEN PRINT " en zoek een goeie accountant"
60 IF INKOMEN >3000 THEN PRINT "...en leent u mij 40 gulden "; NAAM$
61 DAGINKOMEN = INKOMEN/30: PRINT "dat is fl.";DAGINKOMEN;" per dag"
70 FOR n=1 TO 5000:NEXT N:GOTO 5
run
```

Verander in regel 70 het getal 900 in 5000 dat geeft u meer tijd om op het beeldscherm het resultaat te lezen. Het resultaat in regel 61 is onnauwkeurig. U kunt het tot een geheel getal (integer) afronden.
Voer regel 62 in:

```
5 CLS
6 INPUT "Hoe heet u";NAAM$
10 INPUT "Hoe hoog is uw inkomen";INKOMEN
20 IF INKOMEN < 2000 THEN GOTO 30 ELSE 40
30 PRINT "Vraag om opslag":GOTO 70
40 PRINT "Vraag om een grotere auto"
50 IF INKOMEN >4001 THEN PRINT " en zoek een goeie accountant"
60 IF INKOMEN >3000 THEN PRINT "...en leent u mij 40 gulden ";NAAM$
61 DAGINKOMEN=INKOMEN/30: PRINT "dat is fl.";DAGINKOMEN;" per dag"
62 INTEGER.INKOMEN%=DAGINKOMEN: PRINT "of fl.";INTEGER.INKOMEN%;
   " als het niet precies op de cent nauwkeurig hoeft"
70 FOR n=1 TO 5000:NEXT n:GOTO 5
run
```

Als u aan een variabelenaam een soortteken toevoegt, moet u deze in het gehele programma hiermee schrijven, omdat u anders niet dezelfde variabelen aanspreekt. Het soortteken is een deel van een variabelenaam. BASIC regels, die langer zijn dan een beeldschermregel, worden op de volgende regel voortgezet. Dat gebeurt in alle 3 de modi. Gebruikt u MODE 2 zo veel mogelijk voor het schrijven van lange programma´s. Het is gemakkelijker programmaregels te lezen die niet steeds weer op een nieuwe regel beginnen.

Om MODE 2 te krijgen, voert u in:

```
MODE 2
```

Als u met de kleurenmonitor (CTM 640) werkt:

```
INK 1,0
INK 0,13
BORDER 13
```

Het zwarte schrift op de grijze achtergrond is dan beter te lezen. LIST u het programma nu nog eens.

4.8 Formatteren van het beeldscherm

Als u een programma ontwerpt, moet u het van tijd tot tijd herzien. Als eerste kunt u de regelnummers in stappen van 10 hernummeren. Onder de **Ready** melding typt u in:

```
RENUM
```

en LIST dan opnieuw:

```
10 CLS
20 INPUT "Hoe heet u";NAAM$
30 INPUT "Hoe hoog is uw inkomen";INKOMEN
40 IF INKOMEN < 2000 THEN GOTO 50 ELSE 60
50 PRINT "Vraag om opslag":GOTO 110
60 PRINT "Vraag om een grotere auto"
70 IF INKOMEN >4001 THEN PRINT " en zoek een goeie accountant"
80 IF INKOMEN >3000 THEN PRINT "...en leent u mij 40 gulden ";NAAM$
90 DAGINKOMEN=INKOMEN/30:PRINT "dat is fl.";DAGINKOMEN;" per dag"
100 INTEGER.INKOMEN%=DAGINKOMEN:PRINT "of fl.";INTEGER.INKOMEN%;
    " als het niet precies op de cent nauwkeurig hoeft"
110 FOR n=1 TO 5000:NEXT n:GOTO 10
```

Alle regelnummers zijn nu in stappen van 10 georganiseerd. Het belangrijke daarbij is, dat ook de verwijzingen in een programma gewijzigd zijn. Nu moet u de afdrukken op het beeldscherm herzien.

Daarvoor schakelt u ten eerste de tijdlus in regel 110 uit door de instructie REM er in te zetten.

```
10 CLS
20 INPUT "Hoe heet u";NAAM$
30 INPUT "Hoe hoog is uw inkomen";INKOMEN
40 IF INKOMEN < 2000 THEN GOTO 50 ELSE 60
50 PRINT "Vraag om opslag":GOTO 110
60 PRINT "Vraag om een grotere auto"
70 IF INKOMEN >4001 THEN PRINT " en zoek een goeie accountant"
80 IF INKOMEN >3000 THEN PRINT "...en leent u mij 40 gulden ";NAAM$
90 DAGINKOMEN=INKOMEN/30:PRINT "dat is fl.";DAGINKOMEN;" per dag"
100 INTEGER.INKOMEN%=DAGINKOMEN:PRINT" of fl.";INTEGER.INKOMEN%;
    "als het niet precies op de cent nauwkeurig hoeft"
110 REM:FOR n=1 TO 5000:NEXT n:GOTO 10
```

Als men **REM:** aan het begin van de regel zet, wordt de rest van de regel door BASIC overgeslagen. Opmerkingsregels (REM) dienen hoofdzakelijk voor de betere leesbaarheid van een programma. Daar het programma nu na regel 110 stopt en **Ready** meldt, blijft de afdruk op het beeldscherm bewaard. Voer nu eens in:

```
15 mode 1
```

Regel 15 bepaalt nu de modus. Welke mode ook ingesteld was, regel 15 schakelt mode 1 in. De mode instructie wist bovendien het beeldscherm. Regel 10 is daardoor overbodig, maar laat hem toch maar staan.

RUN nu het programma en geef als antwoorden in:

```
Hoe heet u? Hans
Hoe hoog is uw inkomen? 4250
Vraag om een grotere auto
en zoek een goeie accountant
...en leent u mij 40 gulden Hans
dat is fl. 141.666667 per dag
of fl. 142
als het niet precies op de cent nauwkeu
rig hoeft
Ready
```

Dat is niet erg elegant, vooral omdat het woord **nauwkeurig** over twee regels wordt geschreven. Voegt u in:

```
25 PRINT:PRINT
35 PRINT
85 PRINT
```

En verander (EDIT) regel 100 als volgt:

```
100 INTEGER.INKOMEN%=DAGINKOMEN:PRINT "of fl.";INTEGER.INKOMEN%;
    "als het niet precies":PRINT "op de cent nauwkeurig hoeft"
```

Als u het programma nogmaals met RUN uitvoert ziet u dat de tekst beter afgedrukt wordt (in MODE 1). Met regel 120 brengt u de **Ready** mededeling verder naar beneden.

```
120 ?:?:?:?:?
```

Laat u het programma nogmaals RUNnen of voorkom het verschijnen van **Ready** door in te voeren:

```
120 GOTO 120
```

Nu kunt u alleen met [ESC] uit het programma komen. Denkt u eraan, ? is de verkorte vorm van PRINT. LIST u het programma en bekijk het resultaat.

```
10 CLS
15 MODE 1
20 INPUT "Hoe heet u";NAAM$
25 PRINT:PRINT
30 INPUT "Hoe hoog is uw inkomen";INKOMEN
35 PRINT
40 IF INKOMEN < 2000 THEN GOTO 50 ELSE 60
50 PRINT "Vraag om opslag":GOTO 110
60 PRINT "Vraag om een grotere auto"
70 IF INKOMEN >4001 THEN PRINT " en zoek een goeie accountant"
80 IF INKOMEN >3000 THEN PRINT "...en leent u mij 40 gulden ";NAAM$
85 PRINT
90 DAGINKOMEN=INKOMEN/30:PRINT "dat is fl.";DAGINKOMEN;" per dag"
100 INTEGER.INKOMEN%=DAGINKOMEN:PRINT "of fl.";INTEGER.INKOMEN%;
    "als het niet precies ":PRINT "op de cent nauwkeurig hoeft"
110 REM FOR n=1 TO 5000:NEXT n:GOTO 10
120 GOTO 120
```

4.9 LOCATE

De tot nu toe gebruikte BASIC instructies komen overeen met de algemeen gebruikelijke standaard, zoals op de meeste computers aanwezig. LOCATE is een bijzonderheid van Schneider BASIC (en enkele andere BASICs), waarmee de cursor op iedere plaats op het scherm kan worden gezet:

```
LOCATE 10,4
```

brengt de cursor op de vierde regel van boven en de tiende kolom van links. Als u dat in de direkte modus doet, wordt de cursor wel in de goede positie gebracht, maar de READY melding meteen op een nieuwe regel er onder gezet. Voegt u regel 16 toe:

```
16 LOCATE 10,4
run
```

U ziet dat de eerste vraag goed op het scherm staat. De volgende regel wordt niet gewijzigd, daar regel 25 enige spaties invoegt. Drukt u tweemaal [ESC] en voegt u regel 26 toe:

```
26 LOCATE 10,4
```

Als u nu het programma RUNt, worden beide vragen in regel 4 kolom 10 gegeven, waarbij de tweede vraag de eerste overschrijft. Ga nu zelf verder met het op de juiste plaats zetten van alle tekst. U kunt daarvoor de beeldscherm-coordinaten gebruiken die in aanhangsel VI beschreven zijn.

Wilt u alle vragen en commentaren op de zelfde positie afdrukken, dan zou u **CLS:** voor iedere LOCATE moeten invoeren, zodat de nieuwe tekst niet in de resten van vorige teksten komt te staan.

De coordinaten van LOCATE kunnen ook variabelen zijn, gevuld met waarden uit het programma. Het INKOMEN programma heeft nu zijn dienst gedaan. Wilt u het programma behouden, dan legt u het vast op de band (zie hoofdstuk 2). Misschien wilt u het uitbreiden met de instructies, die nu worden besproken.

4.10 IF... THEN

De betekenis en de functie hiervan zijn volledig conform de vertalingen van: INDIEN... DAN. **IF** <logische uitdrukking> **THEN GOTO** <regelnummer> is een van de vele mogelijkheden van deze instructie

IF controleert, of de <logische uitdrukking> waar is. In dit geval wordt de optie uitgevoerd. De IF instructie kan in lussen ingebouwd worden die herhaaldelijk worden uitgevoerd (zogenaamde tijdlussen - recursive loops). RESET de computer met **[CTRL]**, **[SHIFT]** en **[ESC]** en typt u:

```
1 MODE 1
10 SCHNEIDER=0
20 PRINT "SCHNEIDER CPC464 Colour Personal Computer"
30 SCHNEIDER = SCHNEIDER +1
40 IF SCHNEIDER <10 GOTO 20
```

RUNt u dit programma. Het programma blijft in een lus zolang aan de voorwaarde in regel 40 wordt voldaan en keert dus terug naar regel 20. U kunt zien, wat het begrip variabele betekent, want met elke lus verandert de waarde van SCHNEIDER.

Als u wilt zien, hoe de waarde van SCHNEIDER in een programma verandert, kunt u dit met de opdracht in regel 35 uitbreiden.

```
35 LOCATE 1,20: PRINT SCHNEIDER: LOCATE 1,SCHNEIDER +1
run
```

Als dit te snel was, bouw dan een wachtlus in:

```
36 FOR n =1 TO 500:NEXT
```

Nu voegt u als u een CTM640 kleurenmonitor hebt, wat kleur toe:

```
34 BORDER SCHNEIDER
```

Daarmee wordt de kleur door de waarde van de variabele SCHNEIDER bepaald en bij iedere afdruk veranderd. LIST u het programma weer:

```
1 MODE 1
10 SCHNEIDER=0
20 PRINT "SCHNEIDER CPC464 Colour Personal Computer"
30 SCHNEIDER=SCHNEIDER +1
34 BORDER SCHNEIDER
35 LOCATE 1,20: PRINT SCHNEIDER: LOCATE 1,SCHNEIDER +1
36 FOR n =1 TO 500:NEXT
40 IF SCHNEIDER <10 GOTO 20
run
```

U wilt nu natuurlijk alle kleuren zien, en verander daarom regel 40 in:

```
40 IF SCHNEIDER <26 GOTO 20
```

Als u het programma nu laat lopen, ziet u van de 26 mogelijke kleuren, eerst de donkerste en aan het eind de helderste. Met de volgende verandering kunt u de melding een zinvoller vorm geven:

```
35 LOCATE 1,20:PRINT "Border ";SCHNEIDER:LOCATE 1,SCHNEIDER +1
run
```

Omdat u toch met kleuren bezig bent, voert u na het einde van het programma (na de READY melding) in:

```
BORDER 14,6
```

U zult zien, dat de kleur van de beeldrand tussen 14 en 6 wisselt. In het volgende hoofdstuk zult u iets te weten komen over graphics en kleur. Hebt u daarom nog even geduld. Om dit deel af te ronden, voer het volgende in:

ink 1,18,16

dan...

speed ink 1,5

Nu hebt u een pauze verdiend. Vergeet niet het programma op band vast te leggen, als u indruk op uw vrienden wilt maken. RESET de computer.

4.11 ARRAYS (matrices)

Sommige programma's hebben een veelvoud van variabelen nodig om gegevens op te slaan. Dat is eigenlijk geen probleem, maar het kan moeilijk worden greep op de variabelen en hun inhoud te houden. Gelukkig heeft de computer daarvoor een oplossing in de vorm van matrices (ARRAYs).

U zult wel vragen: wat is een matrix?. Dat is een groep van variabelen die allen met de zelfde naam worden aangesproken.

Het beste wordt dit aan de hand van een voorbeeld duidelijk. Denkt u aan een programma, dat een kaartspel simuleert. Daar worden kaarten blindelings getrokken. Dus moet van iedere kaart een lijst worden bijgehouden. Een eenvoudige weg om dit te doen, zou zijn om iedere kaart een variabele te geven en deze een waarde te geven overeenkomstig de positie van die kaart: 1 als ze gespeeld werd en 0 als ze nog niet gespeeld is.

Daaruit volgt dat u 52 variabelen (voor iedere kaart een) nodig hebt, waarbij u zich moet herinneren, welke variabele bij welke kaart behoort. Hier begint nu het nut van een matrix.

Als eerste geeft u de matrix een naam, laten we zeggen STAPEL. Om een element van een matrix vast te leggen, geeft u hem eenvoudig een nummer. Als de eerste 13 elementen harten vertegenwoordigen, dan zou hartenaas gelijk zijn aan STAPEL(1), koning gelijk aan STAPEL(2) en harten 6 gelijk aan STAPEL(9). Ziet u het principe ?.

U kunt echter niet onbegrensd variabelen definieren. U moet de computer een mogelijkheid geven, ruimte in het geheugen vrij te maken voor variabelen. Dit gebeurt met de instructie **DIM**.

Met deze instructie wordt de DIMensionering van een matrix vastgelegd. In ons voorbeeld zijn dat 4 keer 13 kaarten: dus 52 variabelen.

De instructie ziet er dan zo uit:

DIM STAPEL(52)

Daarmee weet de computer dat er plaats voor 52 elementen in het geheugen gereserveerd moet worden (in werkelijkheid worden er 53 gereserveerd omdat ook element 0 erbij hoort).

U kunt nu met de voorbereiding voor een subroutine voor het kaartspel beginnen:

```
10 DIM STAPEL(52)
20 FOR x=1 TO 52
30 LET STAPEL(x)=0
40 NEXT x
............
1000 KAARTEN = INT(RND*52)+1
1010 IF STAPEL(KAARTEN) = 1 THEN GOTO 1000
1020 STAPEL(KAARTEN) = 1
1030 RETURN
```

Let er wel op dat de DIM instructie de eerste regel van een programma is. Dit is noodzakelijk, omdat een matrix maar een keer in een programma gedefinieerd en niet geherdefinieerd kan worden.

In de regels 20 tot 40 wordt aan de elementen van de matrix de waarde 0 toegewezen. De subroutine die in regel 1000 start zoekt blindelings een kaart uit en controleert of ze al gespeeld werd. De waarde van dit element wordt dan op 1 gezet, om aan te tonen, dat de kaart gespeeld is. Dan wordt de subroutine met RETURN verlaten en gaan we in het hoofdprogramma verder.

De matrixverwerking is niet tot eendimensionale matrixen beperkt. Zolang er plaats is in het geheugen, kunt u zoveel plaatsen gebruiken als u wilt. Een driedimensionale matrix met 10 elementen per dimensie wordt zo gedefinieerd:

```
DIM TABEL(10,10,10)
```

Deze techniek is nuttig om grote voorraden gegevens in kleinere overzichtelijke eenheden te verdelen.In ons voorbeeld met het kaart-spel kan men ook 4 eenheden met elk 13 elementen definieren, die dan elk appart toegankelijk zijn. Dan kan men iedere kaart aan de hand van hun kleur bepalen:

```
DIM STAPEL(4,13)
```

Als u nu bijv. klaveren 4 zoekt (in de eerste matrix zou dit element **49** kunnen zijn), dan moet u nu in STAPEL(2,4) zoeken, vooropgesteld dat klaveren als tweede rij van de matrix gedefinieerd is.

In matrixen kunnen getallen of strings worden opgeslagen, bijv. de namen van de bezoekers in een theater per plaats en rij.

4.12 DATA instructie

Deze instructie kan samen met de instructie READ gebruikt worden om gegevens in een programma te brengen. De benodigde gegevens staan in programmaregels door komma's gescheiden en door een DATA instructie aan het begin gedefinieerd. Nu kunnen deze gegevens sequentieel, dit wil zeggen na elkaar, met de READ instructie gelezen worden.

Een voorbeeld daarvan is:

```
10 READ X,Y,Z
20 PRINT X;"+";Y;"+";Z;" = ";X+Y+Z
30 DATA 1,3,5
```

De gegevens kunnen getallen, algebraische uitdrukkingen, strings of een combinatie hiervan zijn. Als niet alle tekens op een regel passen, schrijft u eenvoudig een nieuwe, weer met het woord DATA in het begin. Als de computer een READ instructie vindt, doorzoekt hij het gehele programma naar de eerst volgende gegevens, waar ze ook staan. Zorgt u er voor dat bij iedere READ instructie gegevens aanwezig zijn, anders ontstaan er fouten.

De enige mogelijkheid, de leesvolgorde van READ te wijzigen is met RESTORE. Daarmee wordt de data-pointer aan het begin van het programma, of op het eerste DATAgegeven van de door RESTORE aangegeven regel gezet. Zo kan men deze gegevens indien nodig meerdere keren lezen. het volgende programma demonstreert het gebruik van DATA, READ en RESTORE:

```
10 FOR c=1 TO 6
20 READ X$
30 PRINT X$;" ";
40 NEXT c
50 RESTORE
60 GOTO 10
70 DATA Hallo,hoe,gaat,het,met,u,vandaag
```

Met [ESC] onderbreekt u dit programma.

Let er wel op dat de DATA instructie hier aan het einde van een programma staat, ze kan echter op elke plaats staan.

De DATA instructie moet u niet alleen gebruiken om gegevens te lezen, die moeten worden gePRINT. Nummerieke gegevens kunnen bijv. ook in een SOUND instructie worden ingelezen. Voert u in:

```
10 FOR n=1 TO 30
20 READ s
30 SOUND 1,s,40,5
40 NEXT n
50 DATA 100,90,100,110,120,110,100,0
60 DATA 130,120,110,0,120,110,100,0
70 DATA 100,90,100,110,120,110,100,0
80 DATA 130,0,100,0,120,150
```

Als u niets hoort stelt u dan de geluidssterkte met de rechts van de computer aangebrachte regelaar in.

Ter afsluiting van deze korte inleiding in BASIC, is hier een programma, waarmee u kunt 21-en (in het engels:Pontoon). Het demonstreert het gebruik van vele BASIC functies. Vanwege de passende variabelennamen moet het makkelijk te begrijpen zijn. U kunt creatief zijn en graphics en geluidsfuncties toevoegen en zo van dit BASIC programma een meesterwerk maken. Het doel van dit spel is om zo dicht mogelijk bij 21 te komen. Zou u over de 21 gaan, dan hebt u verloren. Als u het programma intikt, voer dan na regel 1 de instructie AUTO in, dit genereert automatisch de volgende regelnummers.

```
1 REM PONTOON
10 REM INITIALISEREN
20 YC=2:CC=2
30 AAS=0
40 CASSE=0
50 S=0
60 T=0
70 DIM KLEUR$(4)
80 KLEUR(1)="RUITEN"
90 KLEUR(2)="HARTEN"
100 KLEUR(3)="SCHOPPEN"
110 KLEUR(4)="KLAVEREN"
120 CLS
130 DIM STAPEL(52)
140 FOR X=1 TO 52
150 STAPEL(X)=0
160 NEXT X
170 REM geef iedere speler twee kaarten
180 LOCATE 10,3
190 PRINT"U";SPC(15)"BANK"
200 LOCATE 3,5
210 GOSUB 740
220 S=S+F
230 IF F=11 THEN AAS=AAS+1
240 LOCATE 3,6
250 GOSUB 740
260 S=S+F
270 IF F=11 THEN AAS=AAS+1
280 LOCATE 24,5
290 GOSUB 740
300 T=T+F
310 IF F=11 THEN CASSE=CASSE+1
320 LOCATE 24,6
330 GOSUB 740
340 T=T+F
350 IF F=11 THEN CASSE=CASSE+1
360 REM INVOER:KAART NEMEN (N) OF PASSEN (P)
370 X$=INKEY$:IF X$<>"n" AND X$<>"p" THEN 370
380 IF X$="p" THEN 560
390 locate 3,YC+5
400 YC=YC+1
```

```
410 GOSUB 740
420 S=S+F
430 IF F=11 THEN AAS=AAS+1
440 REM PUNTEN EN AAS
450 IF S<22 THEN 370
460 IF AAS=0 THEN 500
470 AAS=AAS-1
480 S=S-10
490 GOTO 450
500 LOCATE 12,19
510 PRINT"U hebt verloren"
520 PRINT:PRINT"Nog een keer (J/N)?"
530 X$=INKEY$:IF X$<>="j" AND X$<>"n" THEN GOTO 530
540 IF X$="j" THEN RUN
550 END
560 IF T>16 THEN 700
570 CC=CC+1
580 LOCATE 24,CC+4
590 GOSUB 740
600 T=T+F
610 IF F=11 THEN CASSE=CASSE-1
620 IF T<21 THEN 560
630 IF CASSE=0 THEN 670
640 CASSE=CASSE-1
650 T=T-10
660 GOTO 620
670 LOCATE 12,19
680 PRINT"U hebt gewonnen!"
690 GOTO 520
700 LOCATE 12,19
710 IF T<S THEN 680
720 PRINT"De bank heeft gewonnen!"
730 GOTO 520
740 REM KAART GEVEN
750 LET KAART=INT(RND(1)*52+1)
760 IF STAPEL(KAART)=1 THEN 750
770 STAPEL(KAART)=1
780 F=KAART-13*INT(KAART/13)
790 IF F=0 then F=13
800 IF F=1 OR F>10 THEN 850
810 PRINT F ;" van ";
820 IF F>10 THEN F=10
830 PRINT KLEUR$(INT((KAART-1)/13)+1)
840 RETURN
850 IF F=11 THEN PRINT"BOER VAN "
860 IF F=12 THEN PRINT"DAME VAN "
870 IF F=13 THEN PRINT"KONING VAN "
880 IF F<>1 THEN GOTO 820
890 F=11
900 PRINT "AAS VAN ";
910 GOTO 830
```

Geef N in voor 'nemen' (de volgende kaart die u krijgt), of P om te passen. Wij beweren niet dat dit het beste programma voor de CPC464 is. Het is een goede grondslag, die u met graphics en muziek verder kunt uitbouwen.

4.13 Logische uitdrukkingen

Een fundamenteel verschil tussen een computer en een rekenmachine zijn de mogelijkheden van de computer, logische uitdrukkingen zoals IF.....THEN te kunnen verwerken. De aard en wijze hoe dit gebeurt, is in deze korte definitie moeilijk te beschrijven. De gegevens, die met deze logische uitdrukkingen verwerkt worden, worden intern als Bit-patronen voorgesteld en werken met de individuele bit. De beschrijving en gebruikt van, is ...logisch maar het is bekend hoe moeilijk logica simpel te beschrijven valt zonder precieze en beknopte definities.

De twee helften van een logische uitdrukking worden argumenten genoemd. Een logische uitdrukking bevat:

<argument>[<vergelijkingsoperator><argument>]

waarbij

<argument> is: NIET <argument>
of :<nummerieke uitdrukking>
of :<relatieve voorwaarde>
of :(<logische uitdrukking>)

Beide argumenten voor vergelijkingsoperatoren moeten hele getallen zijn. ERROR 6 wordt getoond, als dit niet het geval is.

De logische operatoren in de volgorde van verwerking en uitwerking op iedere bit zijn:

AND	resultaat is 0 als niet beide argumentenbits 1 zijn
OR	" " 1 " " " " " 0 "
XOR	" " 1 " " " " " gelijk zijn

AND is de meest gebruikte operator en betekent niet hetzelfde als 'optellen'

PRINT 10 AND 10

Voert tot het resultaat 10

PRINT 10 AND 12

Voert tot het resultaat 8

PRINT 10 AND 1000

Voert tot het resultaat 8

Dat komt daardoor, dat beide getallen, 10 en 1000, in hun binaire omschrijving omgezet worden:

```
      1010
1111101000
```

Dan wordt bit voor bit vergeleken. Als in beide getallen op dezelfde positie een 1 gevonden wordt, is het antwoord 1:

```
0000001000
```

Weer in decimale vorm gezet, is het resultaat weer 8. Dat betekent niets anders, dan dat AND controleert, of 2 voorwaarden gelijktijdig aanwezig zijn. Hier heeft u een zelf verklarend voorbeeld:

```
10 CLS:INPUT "Hoeveelste dag van de maand is het vandaag";DAG
20 INPUT "Hoeveelste maand";MAAND
30 IF DAG=25 AND MAAND=12 GOTO 50
40 GOTO 10
50 PRINT "Vrolijk kerstfeest !"
```

OR vergelijkt ook bits. Het resultaat is altijd 1, behalve als beide bits 0 zijn. In dit geval is het resultaat 0. Nemen we dezelfde getallen als bij het AND voorbeeld:

```
PRINT 1000 OR 10
 1002
```

bitsgewijs:

```
      1010
1111101000
```

Het resultaat is:

```
1111100010
```

En in het programmavoorbeeld:

```
10 CLS
20 INPUT "Hoeveelste maand";MAAND
30 IF MAAND=12 OR MAAND=1 OR MAAND=2 GOTO 50
40 GOTO 10
50 PRINT "Nu is het winter !"
```

```
10 CLS
20 INPUT "Hoeveelste maand";MAAND
30 IF NOT(maand=6 OR maand=7 OR maand=8) GOTO 50
40 CLS:GOTO 10
50 PRINT "Nu is het geen zomer !"
```

Ziet u, dat u meerdere logische voorwaarden met elkaar kunt verbinden, om tot een duidelijke beslissing te komen (tot maximale regellengte):

```
10 CLS:INPUT "Hoeveelste dag van de maand is het vandaag";DAG
20 INPUT "Hoeveelste maand";MAAND
30 IF NOT(maand=12 OR maand=1) AND DAG=29 GOTO 50
40 CLS:GOTO 10
50 PRINT "Het is noch december noch januari, maar het kan een
   schrikkeljaar zijn !"
```

Het resultaat van een vergelijkingsuitdrukking is of -1 of 0. De bit-beschrijving voor -1: alle bits van het hele getal zijn 1; voor 0 zijn alle bits 0. Het resultaat van een logische vergelijkings operatie bij twee van zulke argumenten geeft -1 voor ´waar´ (TRUE) of 0 voor ´fout´ (FALSE)

Controleer dit door in het programma regel 60 toe te voegen:

```
60 PRINT NOT(maand=12 OR maand=1)
70 PRINT (maand=12 OR maand=1)
```

Als het programma loopt geeft u 29 voor de dag en 5 voor de maand. Dan wordt het bericht uit regel 50 getoond en de resultaten van het vergelijken worden in regel 60 en 70 getoond.

Tenslotte is het resultaat van XOR (eXclusieve OR) goed, als beide argumenten verschillend zijn. De volgende matrix toont alle mogelijkheden, die bij logische vergelijkingen bestaan. Dit is een goede demonstratie van datgene, wat bij bitsgewijze logische operaties gebeurt.

Argument A	**1**	**0**	**1**	**0**
Argument B	**0**	**1**	**1**	**0**
AND resultaat	**0**	**0**	**1**	**0**
OR resultaat	**1**	**1**	**1**	**0**
XOR resultaat	**1**	**1**	**0**	**0**

5 GRAPHICS inleiding

KLEUREN EN GRAPHICS VAN DE CPC464

Onderwerpen in dit hoofdstuk:

* BEELDSCHERM-MODI en BEELDPUNTEN
* KLEUREN
* INK, PAPER en PEN
* HET TEKENEN VAN LIJNEN
* VENSTERS (WINDOWs)

5.1 Specifieke eigenschappen

Schneider BASIC voldoet voor het grootste deel aan de industrie-standaard. De meeste instructies zijn zonder grote veranderingen te vertalen naar andere versies van BASIC. De instructies voor graphics, kleur en de tekstcursor zijn speciaal ingericht op het besturen van het beeldscherm via de CPC464 hardware. Alleen als u deze instructies goed begrijpt kunt u alle mogelijkheden van de CPC464 benutten.

Zoals gewoonlijk zijn de BASIC woorden geschreven, zoals ze op het beeldscherm gebruikt worden. Voorbeelden en beschrijving van BASIC woorden en hun functies vindt u in hoofdstuk 8.

5.1.1 Kleur

Zwart is geen kleur of helderheid maar wordt in het hierna volgende altijd als kleur beschouwd als het om de beschrijving van kleur-attributen en de verschillende instructies gaat, die zich met kleur bezig houden.

BORDER, de beeldrand, kan onafhankelijk van de beeldscherm modi op ieder kleurenpaar ingesteld worden. Een mode instructie heeft geen uitwerking op BORDER. Het kan in 2 kleuren wisselen of 1 kleur hebben.

Het aantal beschikbare kleuren (INK), die gelijktijdig gebruikt kunnen worden, hangt af van de beeldscherm modus, die met MODE bepaald wordt. INK kan wisselen in twee kleuren of met een kleur gebruikt worden. De kleur van de ondergrond PAPER, PEN en de graphics PEN kan uit het beschikbare INK palet gekozen worden.

5.1.2 Transparant optie en de relaties tot PEN, INK, en PAPER

Behalve wanneer wisselende kleuren gekozen zijn, worden 2 kleuren (INK) voor de tekst van het beeldscherm gebruikt. Een INK bepaalt de kleur van PEN, een andere PAPER (de ondergrond).
Let op: Het getal in de PAPER instructie geeft de INK aan, die voor dit getal gedefinieerd is en NIET het nummer van de kleur, zoals in aanhangsel IV aangegeven is. Het zelfde gebeurt met de PEN instructie. Ook daar definieert het getal de INK gedefinieerd voor dat PEN nummer en niet het nummer van de kleur, zoals in aanhangsel IV beschreven is.

Als geen nummer voor PAPER ingevoerd wordt, is de standaardwaarde 0, terwijl bij PEN zonder nummervermelding de standaardwaarde 1 is. Om INK op groen voor PAPER 0 te zetten, kleurnummer 9, geeft u de volgende opdracht in:

```
INK 0,9
```

Evenzo als u PEN 1 van de kleur zwart wilt voorzien:

```
INK 1,0
```

Als u PAPER van dezelfde INK als PEN voorziet, bijv. INK 0,0 dan wordt het gehele beeldscherm zwart.
De tekst kan transparant gemaakt worden (de cursor is altijd transparant) of ondoorzichtig. Gebruikt u daarvoor de besturings-tekens, die de mogelijkheden van de BASIC graphicsinstructies nog aanzienlijk verruimen. In de transparant modus kunt u of de PAPER kleur vergeten en de graphics overschrijven of de hele ondergrond (PAPER) overschrijven. Dit korte voorbeeldprogramma demonstreert de mogelijkheden:

```
[CTRL][SHIFT][ESC]
10 MODE 1
20 INK 2,19
30 DRAW 200,200,2
40 LOCATE 1,21
50 PRINT "1 normaal"
60 PRINT CHR$(22)+CHR$(1)
70 ORIGIN 0,0
80 DRAW 500,200,2
90 LOCATE 12,18
100 PRINT "2 transparant"
110 PRINT CHR$(22)+CHR$(0)
120 LOCATE 22,15
130 PRINT "3 weer normaal"
```

De eerste DRAW instructie in regel 30 wordt uitgevoerd, voordat de transparantmodus door de instructie **CHR$(22)+CHR$(1)** in regel 60 geactiveerd wordt. U ziet hoe de overlappende punten in kleur veranderd worden, en hoe in de transparantmodus de INK, die de karakters inkleurt compleet overlapt worden.

Verwisselt u de regels 60 en 110 en zie hoe de afbeelding verandert. U vindt in aanhangsel VI de complete lijst van de aanvullende instructies.

5.2 Beeldscherm-modi

Er zijn 3 modi voor het beeldscherm (tekst en graphics).

a) Normaal

Mode 1: 40 kolommen x 25 regels, 4 x INK voor tekst,
320 * 200 beeldpunten, ieder afzonderlijk te definieren in een van de 4 kleuren

b) Meerkleuren

Mode 0: 20 kolommen x 25 regels, 16 x INK voor tekst,
160 * 200 beeldpunten, ieder afzonderlijk te definieren in een van de 16 kleuren

c) High resolution

Mode 2: 80 kolommen x 25 regels, 2 x INK voor tekst,
640 * 200 beeldpunten, ieder afzonderlijk te definieren in een van de 2 kleuren

Zoals u kunt zien ligt het verschil in het aantal horizontale beeldscherm-elementen. Laat u zich niet door de mogelijke strepen op het beeldscherm die u soms op een TV kunt zien, verwarren. Dat is normaal.

Ieder van de verschillende beeldschermmogelijkheden wordt DISPLAY MODE genoemd. Een kan er telkens actief zijn. Als u van ene modus op het andere omschakelt, wordt het hele beeldscherm inclusief alle tekst en graphics vensters uitgewist (zie CLS en CLG). Het programma in het geheugen wordt niet beinvloed.

U kunt de modi direkt of in een programma veranderen.

5.2.1 MODE 0 meer kleuren graphics mode

16 van de 27 kleuren kunnen gelijktijdig getoond worden. Ieder beeldscherm element kan met een kleur geprogrammeerd worden. Iedere regel heeft 160, iedere kolom heeft 200 beeldpunten. In aanhangsel VI vindt u een plan van het beeldschermraster.
In MODE 0 kunt u 20 tekens op elke van de 25 regels schrijven.

5.2.2 MODE 1 normaal

Als u de CPC464 inschakelt, is mode 1 geactiveerd. 4 uit 27 kleuren kunnen gelijktijdig getoond worden. U kunt, als u wilt zeer snel tussen de 27 kleuren omschakelen. Iedere regel heeft 320, iedere kolom heeft 200 beeldpunten. Ook dit beeldschermraster vindt u in aanhangsel VI.
In mode 1 staan 40 tekens op elke van de 25 regels ter beschikking.

5.2.3 MODE 2 High resolution

In mode 2 kunnen gelijktijdig 2 kleuren getoond worden. Het wordt hoofdzakelijk gebruikt als men 80 tekens tekst per regel produceren wil. Daarmee is deze modus speciaal geschikt voor het vervaardigen van programma's, omdat men hier een veel groter programmafragment kan zien.

Mode 2 levert per regel 640 en per kolom 200 beeldpunten.

5.2.4 Probeert u dit....

RESET de CPC464 met [CTRL][SHIFT][ESC]. Voer dit programma in:

```
5 REM GRAPHICS VOORBEELD
10 MODE 1
16 INK 3,6: REM zet kleur voor DRAW in 90
17 BORDER 1: REM donkerblauw
20 CLG: REM wis het beeldscherm
30 b%=RND*5+1 :REM zet de pseudo random integervariabele
40 c%=RND*5+1
50 ORIGIN 320,200: REM leg het grafiesch startpunt vast
60 FOR a=0 TO 1000 STEP PI/30
70 x%=100*COS(a)
80 MOVE x%,x%: REM verschuif de graphics cursor
90 DRAW 200*COS(a/b%),200*SIN(a/c%),3: REM teken een lijn
91 IF INKEY$<>"" THEN 20
100 NEXT: REM terug naar 60 indien geen onderbreking in 91
110 GOTO 20
```

Met RUN start u het programma. Druk op een of andere toets als u een nieuw beeld wilt krijgen. Dit programma toont enige belangrijke eigenschappen van de hard-en software van de CPC464: de CPC464 'beschrijft' het beeldscherm zeer glad en gelijkmatig zonder knipperen en de software kent instructies, waarmee op eenvoudige wijze mooie effekten kunnen worden bereikt. De aanwijzing REM (opmerking, REMark) geeft u de mogelijkheid, uw programma te documenteren. Daarmee kunnen u en anderen begrijpen hoe het programma werkt.
Aan de hand van sommige regelnummers kan men herkennen, welke instructies later ingevoegd worden. Met RENUM worden de regels opnieuw in stappen van 10 genummerd.
U kunt ook de originele nummering zo laten, om te zien hoe het programma stapsgewijs ontstaan is.

Slaat u nu het programma met de instructie SAVE op de cassette op. Bijvoorbeeld SAVE"graphics 5.5.84"

Dit voorbeeld tekent een meerkleuren beeld:

```
10 a$=INKEY$:REM DRUK EEN WILLEKEURIGE TOETS VOOR EEN NIEUW PATROON
20 IF a$="" THEN 10
30 CLS
40 m=INT(RND*3):REM SELECTEERT TOEVALSGETAL TUSSEN 0 EN 2
50 IF m>2 THEN 40:REM OPNIEUW ALS WAARDE GROTER IS DAN 2
60 MODE m
70 i1=RND*26:REM SELECTEER TOEVALS INK WAARDE
80 i2=RND*26
90 IF ABS(i1-i2)<5 THEN 70
100 INK 0,i1:INK 1,i2
110 s=RND*5+3
120 ORIGIN 320,-100
130 FOR x=-1000 TO 0 STEP s
140 MOVE 0,0
150 DRAW x,300:DRAW 0,600
160 MOVE 0,0
170 DRAW -x,300:DRAW0,600
180 a$=INKEY$
190 IF a$<>"" THEN 30:REM ONDERBREKEN VAN DE LUS DOOR DRUKKEN VAN
    WILLEKEURIGE TOETS
200 NEXT x
210 GOTO 10
```

Dit en het vorige voorbeeld tonen wiskundige begrippen op kleurige en duidelijke wijze. Beide programma´s doen eigenlijk hetzelfde.

Door een toevalsgenerator (RaNDom generator) worden willekeurige getallen geproduceerd die na de juiste behandeling verschillende lijntekeningen op het beeldscherm projecteren.

Uw CPC464 is een ideale electronische tekentafel. Een van de klassieke geometrische modellen is de SINUS curve:

```
10 REM sinus curve tekenen
20 MODE 2
30 INK 1,21
40 INK 0,0
50 CLS
60 DEG
70 ORIGIN 0,200
80 FOR n=0 TO 720
90 y=SIN(n)
100 PLOT n*640/720,198*y,1
110 NEXT
```

De instructie PLOT in regel 100 tekent de lijn punt voor punt na iedere berekening in de FOR...NEXT lus (regel 80-110)

De CPC464 heeft veel eenvoudige en krachtige instructies. U kunt het effekt van dit programma verhogen als u het volgende toevoegt:

```
15 BORDER 6,9
```

Laat dit programma opnieuw RUNnen. De beeldrand (BORDER) wisselt nu tussen de kleuren 6 en 9. De snelheid van het wisselen is door de standaardwaarde vastgelegd. Zodra het programma onderbroken wordt in in de lus doordat u [ESC] drukt (1x =halt, 2x =stop), voegt u de volgende regel toe:

```
120 GOTO 50
```

De beeldrand wisselt zin kleur nog steeds, ook als het programma gestopt is, omdat de randkleur onafhankelijk van het programma gestuurd wordt. U kunt het knipperen van het programma stoppen en de kleur in helderblauw veranderen, als u tweemaal op [ESC] drukt en regel 15 als volgt verandert:

```
15 BORDER 2
```

Als u nu het programma uitvoert, stopt het knipperen. U kunt de kleur van de curve en de achtergrond veranderen, als u de kleuren van INK in regel 30 en 40 verandert. Met LIST moet u nu de volgende programma-LISTing krijgen:

```
10 REM sinus curve tekenen
15 BORDER 2
20 MODE 2
30 INK 1,2
40 INK 0,20
50 CLS
60 DEG
70 ORIGIN 0,200
80 FOR n=0 TO 720
90 y=SIN(n)
100 PLOT n*640/720,198*y,1
110 NEXT
120 GOTO 50
```

Het getal 1 aan het einde van de PLOT instructie draagt de computer op, de curve in de kleur van INK 1 te tekenen. Leest u hoofdstuk 8 op de precieze opbouw en de functie van de PLOT instructie na.

Als u de curve die op het beeldscherm getoond wordt, precies bekijkt, zult u bemerken dat het geen doorgetrokken lijn is, maar allemaal kleine punten die tegen elkaar getekend zijn. Deze punten zijn de beeldpunten (PIXELs), die al eerder besproken werden.

5.2.5 De graphics-cursor en het tekenen van lijnen

U heeft nu op verschillende wijzen geprobeerd een graphics-programma te maken. U kon enige instructies testen. Bij het tekenen van lijnen op het beeldscherm moet u met enkele.belangrijke punten rekening houden teneinde niet in de war te raken.

Het eerste punt waar u rekening mee moet houden is de actuele toestand van het programma in het geheugen. De kleuren blijven actief, ook als het programma beeindigd is. Zelfs wanneer u met NEW een nieuw programma begint, of alleen daarmee het oude programma uitveegt, heeft dat geen uitwerking op de aangewezen kleuren. Alleen wanneer u de computer RESET, heeft u een volkomen leeg geheugen. (alvorens dit te doen moet u alle gegevens die kunt gebruiken SAVEn).
Overtuigt u zich nu hiervan dan kunnen we verder gaan:

```
NEW:CLS
```

Nadat u het vorige programma afgebroken heeft, toetst u in:

```
DRAW 100,100
```

De DRAW instructie trekt een lijn van de laatste graphics cursor-positie naar de X,Y coordinaten van de DRAW instructie (100,100). De graphics-cursor is onzichtbaar en staat altijd op de positie waar de graphics-instructie iets tekent.
Als u een positie wilt vinden moet u de functies X-POS en Y-POS gebruiken. Toetst u nu in:

```
PRINT XPOS
```

het antwoord is

```
100
```

(hetzelfde geldt voor de Y-POS)

Als de tekst de laatste beeldschermregel overschrijdt en daardoor het beeld naar boven gerold wordt, rolt ook de graphics naar boven, maar de graphics-cursor behoudt zijn positie. Probeert u het. Drukt u op de pijl-naar-beneden totdat het beeld naar boven verdwenen is. Test u de waarde van X-POS en Y-POS. Er is niets veranderd.
Wilt u in een bepaalde kleur tekenen, dan moet u de opdracht aan het einde van de DRAW instructie toevoegen (net zoals bij de PLOT instructie op de vorige bladzijde).
Als eerste moet u echter INK gespecificeerd hebben. INK is beperkt door de modi van het beeldscherm, dat op dat moment actief is. Het volgende programma laat dat zien:

```
10 MODE 1
20 INK 0,10
30 ORIGIN 0,0
40 INK 1,26
50 INK 2,0
60 DRAW 320,400,1
70 DRAW 640,0,2
```

Het volgende programma demonstreert alle mogelijkheden die tot nu toe besproken werden en toont daarnaast nog een paar andere nuttige mogelijkheden.
In regel 10 worden INK en kleur zo gedefinieerd, dat alle mogelijke resten van het vorige programma gewist zijn en het nieuwe programma de gewenste resultaten levert:

```
10 INK 0,0:INK 1,26:INK 2,6:INK 3,18: BORDER 0
20 REM dit programma tekent patronen
30 MODE 1:DEG
40 PRINT "3,4 of 6 kantig patroon ? ";
50 LINE INPUT p$
60 IF p$="3" THEN sa=120:GOTO 100
70 IF p$="4" THEN sa=135:GOTO 100
80 IF p$="6" THEN sa=150:GOTO 100
90 GOTO 50
100 PRINT:PRINT "berekenen";
105 IF p$="3" THEN ORIGIN 0,-50,0,640,0,400 ELSE ORIGIN 0,0,0,640,0,400
110 DIM cx(5),cy(5),r(5),lc(5)
120 DIM np(5)
130 DIM px%(5,81),py%(5,81)
140 st=1
150 cx(1)=320:cy(1)=200:r(1)=80
160 FOR st=1 TO 4
170 r(st+1)=r(st)/2
180 NEXT st
190 FOR st=1 TO 5
200 lc(st)=0:np(st)=0
210 np(st)=np(st)+1
220 px%(st,np(st))=r(st)*SIN(lc(st))
230 py%(st,np(st))=r(st)*COS(lc(st))
240 lc(st)=lc(st)+360/r(st)
245 IF (lc(st) MOD 60)=0 THEN PRINT ".";
250 IF lc(st) < 360 THEN 210
252 px%(st,np(st)+1)=px%(st,1)
254 py%(st,np(st)+1)=py%(st,1)
260 NEXT st
265 CLS:INK 1,2
270 st=1
280 GOSUB 340
290 LOCATE 1,1
300 EVERY 25,1 GOSUB 510
310 EVERY 15,2 GOSUB 550
320 EVERY 5,3 GOSUB 590
330 GOTO 330
340 REM tekent cirkel met 3,4 of 6 andere er omheen
350 cx%=cx(st):cy%=cy(st):lc(st)=0
360 FOR x%=1 TO np(st)
370 MOVE cx%,cy%
380 DRAW cx%+px%(st,x%),cy%+py%(st,x%),1+(st MOD 3)
390 DRAW cx%+px%(st,x%+1),cy%+py%(st,x%+1),1+(st MOD 3)
400 NEXT x%
```

```
400 NEXT x%
410 IF st=5 THEN RETURN
420 lc(st)=0
430 cx(st+1)=cx(st)+1.5*r(st)*SIN(sa+lc(st))
440 cy(st+1)=cy(st)+1.5*r(st)*COS(sa+lc(st))
450 st=st+1
460 GOSUB 340
470 st=st-1
480 lc(st)=lc(st)+2*sa
490 IF (lc(st) MOD 360)<>0 THEN 430
500 RETURN
510 ik(1)=1+RND*25
520 IF ik(1)=ik(2) OR ik(1)=ik(3) THEN 510
530 INK 1,ik(1)
540 RETURN
550 ik(2)=1+RND*25
560 IF ik(2)=ik(1) OR ik(2)=ik(3) THEN 550
570 INK 2,ik(2)
580 RETURN
590 ik(3)=1+RND*25
600 IF ik(3)=ik(1) OR ik(3)=ik(2) THEN 590
610 INK 3,ik(3)
620 RETURN
```

Bij het RUNnen van dit programma wordt u in regel 40 om een invoer gevraagd. Antwoordt u met 3. Dat geeft het snelste resultaat. Dan verschijnt het bericht "Ik reken" en om de paar seconden (regel 245) wordt een punt getoond, die erop wijst, dat het programma nog ´rekent´.

In de regels 300 tot 320 worden subroutines opgeroepen. Daar worden de verschillende kleuren INK aan het knipperen gebracht, bepaald door de EVERY instructies. Als u het geheel langzamer wilt laten verlopen, verander dan de regels 300-320 als volgt:

```
300 EVERY 250,1 GOSUB 510
310 EVERY 150,2 GOSUB 550
320 EVERY  50,3 GOSUB 590
```

De beschrijving van EVERY vindt u in hoofdstuk 8. Het is een van de nuttigste eigenschappen van Schneider BASIC. Kijkt u eens wat er gebeurt, als u het programma met [ESC] laat pauzeren dan een paar seconden later door het indrukken van een of andere toets weer start.

Het programma probeert de verloren tijd in te halen. Het beeld knippert zo snel mogelijk, omdat de instructies achter elkaar werden opgeslagen. Daar het programma maar een beperkt geheugen ter beschikking staat, wordt als het geheugen vol is de volgende EVERY instructie genegeerd, tot er weer ruimte is. Ruimte wordt gemaakt, als men het programma verder laat lopen en de opgeslagen instructies afgewerkt kunnen worden.

5.3 Vensters (Windows)

U kunt het beeldscherm verdelen in 1 tot 8 vensters waarin teksten kunnen worden geschreven en in een graphics venster. Vensters worden door het gebruik van de mode instructie geRESET naar hun standaard-waarde. In Hoofdstuk 8 vindt u een nadere beschrijving.
Let op: Als het tekstvenster het hele beeldscherm omvat (dat is de voorinstelling bij het opstarten) zorgt de hardware voor een snel ´doorrollen´. Als het tekstvenster niet het hele beeldscherm omvat neemt de software het rollen over en dat gaat veel langzamer.

De WINDOW instructie specificeert de venstergrenzen van de beeldscherm-´streams´. Vensters kunnen elkaar overlappen. Daarmee heeft u een snelle methode gekleurde rechthoeken te tekenen. Voordat u dit nu onderzoekt tikt u in:

```
KEY 139, "mode 2:paper 0:ink 1,0:ink 0,10:list"+chr$(13)
```

Daarmee wordt de kleine [ENTER] toets als functietoets gedefinieerd, met het doel de tekst in zichtbare kleuren te brengen als u zich met PEN en PAPER eens vergist hebt, doordat u alle twee dezelfde kleur gegeven hebt en daardoor niets meer kunt lezen. Het volgende voorbeeld tekent enige vensters over het hele beeldscherm en toont 2 belangrijke punten:

```
5 MODE 0
10 FOR n=0 TO 7
20 WINDOW #n,n+1,n+6,n+1,n+6
30 PAPER #n,n+4
40 CLS #n
50 FOR c=1 TO 200: NEXT
60 NEXT
```

De eerste: dat ieder nieuw venster een oud overschrijft, de tweede: dat meldingen altijd op stream #0 wordt uitgevoerd, tenzij men uitdrukkelijk iets anders overeenkomt. Voordat u iets anders doet typt u in:

```
LIST
```

Het programma wordt op stream #0 geprint. Probeert u:

```
LIST #5
```

Dan:

```
CLS #6
```

Dit toont u dat de laatst aangeroepen beeldscherm-stream alle andere overschrijft en dat de systeemmelding READY op stream #0 komt, ook als het LISTen naar stream #5 gestuurd werd. Voegt u regel 55 toe om de WINDOW SWAP instructie te gebruiken:

```
55 IF n=3 THEN WINDOW SWAP 7,0
```
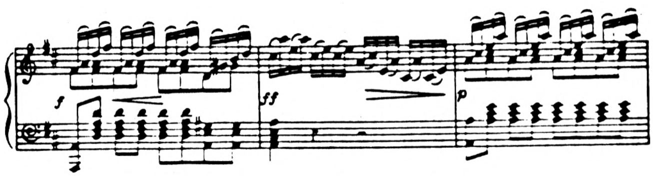
U zult terecht vermoeden dat deze aanwijzing de melding READY naar het venster met het nummer 7 stuurt. RUN en ziet! Met dit eenvoudige programma heeft u een voorstelling gekregen hoe WINDOWs functioneren.

6 MUZIEK-INLEIDING

Geluidseffecten worden door een luidspreker in de computer geproduceerd. Als u de modulator MP1 en uw televisie gebruikt, zet u dan het volume van de televisie zo laag mogelijk.

Het volumeniveau regelt u met de volumeregelaar helemaal rechts op de zijkant van de computer. U kunt het geluid (in stereo) ook over uw stereo-installatie afspelen door het aansluiten van de AUXiliary-aansluiting met de I/O aansluiting links aan de achterkant van de computer. Zo kunt u de muziek die door de computer geproduceerd wordt, over uw Hifi-luidspreker of hoofdtelefoon horen.

Het geluid van de CPC464 is uitstekend. U moet echter de filosofie, die achter de geluidssoftware steekt, begrijpen om tot het beste resultaat te komen.



Onderwerpen in dit hoofdstuk:

* TOONHOOGTE
* MUZIEK-INSTRUCTIES
* VARIATIES
* OPSLAAN EN SYNCHRONISEREN

Als u alleen wilt, dat uw computer een pieptoon voortbrengt, dan voert u in:

```
PRINT CHR$(7)
```

en slaat de rest van het hoofdstuk over. Maar dan zult u de interessantste en een van de mooiste eigenschappen van de CPC464 nooit leren kennen. Dit hoofdstuk geeft eerst een volledig overzicht over de geluidseigenschappen en verklaart dan uitvoerig de BASIC woorden en hun gebruik. Het zal de programmeur aanleren, de volgorde van de geluiden te programmeren, verschillende instrumenten na te bootsen en daarmee melodieen te spelen

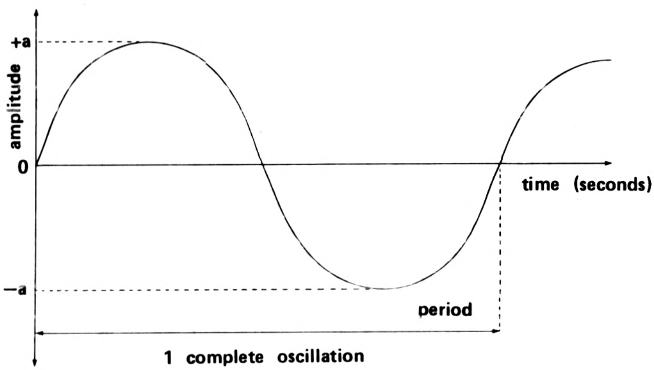
6.1 De beginselen van SOUND

Als u de klank van een noot in een melodie hoort zult u verschillende eigenschappen ontdekken:

1) Toonhoogte en verandering van toonhoogte tijdens de duur van een noot.
2) Geluidssterkte en verandering van geluidssterkte tijdens de duur van een noot.
3) Lengte van een noot.

6.2 Toonhoogte

Bij een noot is de toonhoogte de belangrijkste factor. Een muzieknoot kan als een regelmatige trilling beschreven worden. Alle trillingen hebben een frequentie, een periode en een amplitude (zie afb.1)



Afb.1: De karakteristieken

De frequentie is het aantal trillingen per seconde. De periode is de tijd voor een hele trilling. De amplitude is een eigenschap van de geluidssterkte en speelt hier voor de definitie van de toonhoogte van een noot geen rol.
De verhouding tussen frequentie en periode kan door de volgende formule uitgedrukt worden:

Frequentie (in Hz) =125000/<periode>

Daarmee produceert een <periode> van 1000 een 125 Hz toon en een <periode> van 125 produceert een toon van 1000 Hz (1kHz).

Op beide wijzen zou men de toonhoogte kunnen definieren. Voor Schneider BASIC werd de <toonhoogte> in de vorm van perioden gekozen. Laat u zich niet door het feit verwarren, dat de toonhoogte daalt, als de waarde voor de periode stijgt. Als we verder over toonhoogte spreken dan bedoelen we de periode. Periode is dus NIET een maat voor de tijdsduur van een toon!

De toonhoogte kan in de SOUND-instructie via de parameter<toonhoogte> worden gezet. Het bereik voor toonhoogten gaat van 0 tot 4095 en moet als integer geschreven worden. Dat kan soms tot een afrondings-factor leiden als u een melodie opbouwt, maar zelfs een geschoold oor zal dit bijna niet waarnemen (zie aanhangsel VII)

Als een noot op een echt muziekinstrument gespeeld wordt, kan de toonhoogte veranderen, soms is dit ook opzettelijk (vibrato). Met de ENT-instructie (ENvelope Tone, toonhoogtevariatie) kan een speciale vorm voor de structuur van de toon worden vastgelegd en in de SOUND-instructie worden gebruikt. Deze techniek wordt in het vervolg als <toon-hoogtevariatie> aangeduid.

6.3 Volume

Het volume is de maat voor de sterkte van het geluid. Om de aanvangswaarde van een SOUND instructie te zetten staat u een gebied ter beschikking:

Kiest u uit het bereik 0-15 een getal en wel een integer. Hoe hoger het getal, des te harder is het geluid, de toon. Deze waarde is eveneens een onderdeel van de SOUND instructie en wordt daar als <geluidssterkte> aangegeven. Als u al een paar eenvoudige noten geprobeerd heeft, is u zeker opgevallen hoe eentonig een normale toon klinkt als ze geen extra ´behandeling´ heeft gehad. Dat komt omdat bij een normaal muziekinstrument de toon eerst aanzwelt en dan weer wegsterft.

De verschillende vormen van het aanzwellen en wegsterven voor de verschillende muziekinstrumenten kunnen met de computer door de electronische volumeregeling gesimuleerd worden. Dat gebeurt met de ENV-instructie (ENvelope Volume) en wordt als <geluidssterktevariatie> aangeduid.

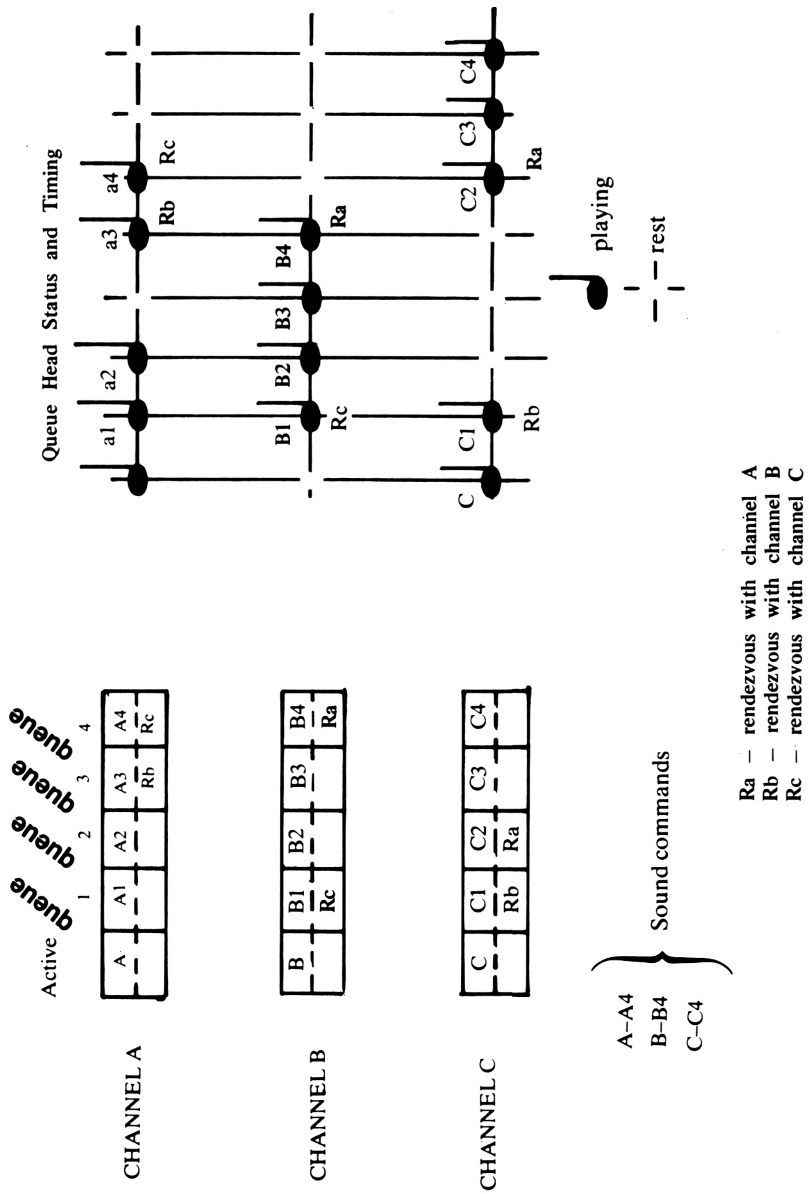
6.4 De lengte van een noot

De lengte van een noot is een wezenlijk bestanddeel van een muziekstuk. De grondeenheid van de muzikale lengte van een noot is de kwart-maat. Er zijn achtste-, halve- en hele noten. Altijd zijn dit delen of meervouden van een hele noot. Melodieen kunnen echter in verschillende snelheden gespeeld worden. Daarom moet de juiste lengte of de tijdsduur voor de kwartnoot gedefinieerd worden.
Dit kan men door een parameter in het SOUND-instructie bereiken, die als <lengte> aangeduid wordt, een integer met de eenheid van 1/100 sec.(bijv.100=1 sec.). Let u erop, dat de <lengte> ook door de lengte van de <geluidssterktevariatie> vastgelegd kan worden. Bedenkt u dit, als u de ENV-instructie gebruikt.

6.5 Andere soorten van tonen

Met de computer kunnen ook andere geluiden (witte ruis) geproduceerd worden, die niets met de oorspronkelijke tonen te maken hebben. Zulke effecten kunnen b.v. als achtergrond van melodieen gebruikt worden om interessante variaties te creeren of helemaal zelfstandig om speciale effecten te bereiken. Een explosie is zo´n geluid dat door de <geluidssterktevariatie> gevormd wordt. Daarbij wordt de frequentie - door een toevals generator gestuurd- rond een <ruisfrequentie> gevarieert, die in de SOUND instructie vastgelegd werd. Daarbij dient opgelet te worden dat weliswaar drie kanalen ter beschikking staan en daarmee tot en met 3 <toonhoogten>, er kan echter slechts een <ruisfrequentie> vastgelegd worden, die dan over de gewenste kanaalcombinatie weergegeven wordt.

Rendezvous for sound channels



6.6 Meertonig geluid en kanalen

De meeste muziek stukken zijn voor minstens twee muzieksleutels geschreven: viool- en bassleutel. Om dit op de CPC464 mogelijk te maken, werden 3 SOUND kanalen(A, B en C) ontworpen. Over alle 3 de kanalen kan onafhankelijk van elkaar gespeeld worden, maar ook samen. De keuze van het kanaal wordt in het SOUND commando met de parameter <kanaal-status> bepaald.

6.7 Kanaal-"Queues" of wachtrijen

Voor ieder SOUND-kanaal is er een queue (wachtrij) van SOUNDs die afgespeeld moeten worden. In iedere wachtrij is plaats voor vijf afzonderlijke SOUND-instructies: een aktief en vier wachtende. Het Operating System van de CPC464 is in staat andere instructies af te werken, terwijl de sound-wachtrij wordt afgespeeld.

6.8 Kanaalstatus

Het BASIC woord **SQ** wordt gebruikt om de status van een kanaal, dat u wilt bekijken, te bepalen. Er wordt informatie over vrije ruimte in de wachtrij, "rendezvous" en stopplaatsen (HOLD) gegeven.
ON SQ GOSUB is een onderbrekingsinstructie, dat gebruikt kan worden om naar het programmadeel, waarin de muziek gemaakt wordt, terug te springen.

6.9 Rendezvous en stopplaatsen

Om de kanalen te synchroniseren staat een zogenaamde **RENDEZVOUS-** techniek ter beschikking. Daarvoor worden in twee of meer kanalen een vlag gezet, die het gelijktijdig verwerken van de SOUND instructie verzorgen. Dat is een zeer nuttige mogelijkheid als bijv, toonreeksen op een kanaal onderbroken worden en op het andere niet.

De instructie HOLD (zie SOUND <kanaal -status>)is belangrijk wanneer algehele kanaalsynchronisatie nodig is. Hierna kan de timing van doorklinkende tonen worden bijgesteld, wanneer er rusten in afzonderlijke kanalen voorkomen. Dan wordt een statement met HOLD en RELEASE gebruikt.

6.10 Instructiereeks voor het genereren van geluid

De instructievorm is: **SOUND G,H,I,J,K,L,M**
waarbij..

G: <kanaal-status>
H: <toon-hoogte>
I: <lengte>
J: <geluidssterkte>
K: <geluidssterkte-variatie>
L: <toonhoogteverloop>
M: <ruis-frequentie>

In de SOUND instructie bestaan alle parameters uit integers. Alleen de eerste twee moeten altijd worden gegeven. Als de overigen niet worden gegeven, gelden de standaard-waarden voor deze parameters. Deze worden bij de beschrijving van de parameters uitgelegd.

6.10.1 PARAMETER-BESCHRIJVING

G: kanaal status

Bereik 1....255

Standaardwaarde niet aangegeven: invoer noodzakelijk.

Met de CPC464 kunnen gelijktijdig 3 verschillende tonen gespeeld worden en wel over de 3 SOUND kanalen A, B en C. De invoer gebeurt als decimale integers, maar als het decimaal getal in een 8 bit waarde wordt omgezet, activeren de gezette bits de volgende opdrachten:

DECIMAAL	BIT	OPDRACHT
1	0 LSB	stuur SOUND naar kanaal A
2	1	stuur SOUND naar kanaal B
4	2	stuur SOUND naar kanaal C
8	3	Rendezvous met kanaal A
16	4	Rendezvous met kanaal B
32	5	Rendezvous met kanaal C
64	6	Hold
128	7 MSB	Flush

LSB = laagste bit (least Significant Bit)
MSB = hoogste bit (Most Significant Bit)

Voorbeelden voor integer invoer:

2 = zendt het er op volgende SOUND naar kanaal B
5 = zendt het er op volgende SOUND naar kanaal A en C
98 = 64+32+2
= zendt het er op volgende SOUND naar kanaal B, RENDEZVOUS met kanaal C en stop (HOLD).

Als een RENDEZVOUS op 2 of meerdere kanalen gearrangeerd moet worden, moet de <kanaal-status> van een van de kanalen op de juiste tijd ´gevlagd´ worden. Bij het zetten van een stop met HOLD stopt het geluid en de toonlus daarachter wordt zolang vastgehouden, tot een RELEASE instructie gegeven wordt, of met een later SOUND instructie leeg gemaakt wordt (FLUSH).

Als de flush-bit voor een kanaal gezet is, wordt de SOUND die daarbij mee wordt ingevoerd, meteen uitgevoerd. De wachtrij (queue) blijft daarbij leeg en het kanaal nonactief. Elke SOUND die juist gespeeld werd ,wordt afgebroken.

H: toon-hoogte

Bereik 0....4095

Standaardwaarde niet gegeven: invoer noodzakelijk.

Legt de toonhoogte en dus de frequentie van een toon vast. De frequentie krijgt men door de volgende formule: frequentie = 125000/toonhoogte.
Als u 0 geeft wordt er geen frequentie bepaald, dat is nuttig wanneer ruis moet worden weergegeven.

I: lengte

Elke integer waarde in het bereik van -32768....+32767 is geoorloofd.

Standaardwaarde indien niets wordt ingevoerd: 20.

Bij waarden groter dan 0 geeft de ingevoerde waarde, de lengte in 1/100 seconden. Als de waarde 0 is, wordt de lengte door de geluidssterkte-variatie bestuurd. Als het getal negatief is, geeft het het aantal herhalingen van de ingestelde geluidssterktevariatie aan.

J: Geluidssterkte

Een integer waarde in het bereik van 0...15

Standaardwaarde, indien niet gegeven: 12

Dit is de geluidssterkte aan het begin van een toon, die door de geluidssterkte-variatie veranderd kan worden. 0= geen volume.

K: geluidssterkte-variatie

Een integer waarde in het bereik van 0...15.

Standaardwaarde, indien niet gegeven: 0.

De waarde specificeert een vooraf gegeven variatie, die met ENV gedefinieerd kan worden. Een vaste definitie is geluidssterkte-variatie nr.0. Deze kan niet veranderd worden en is op 2 seconden op de <volumeschaal> vastgelegd.

L: Toonhoogteverloop

Een integer waarde in het bereik van 0...15.

Standaardwaarde, indien niet gegeven: 0.

De waarde specificeert een vooraf gegeven verloop, die met ENT gedefinieerd kan worden. Een vaste definitie is toonhoogteverloop nr.0, die niet veranderd kan worden en op die toonhoogte vastgezet is.

M: Toonhoogte van de ruis

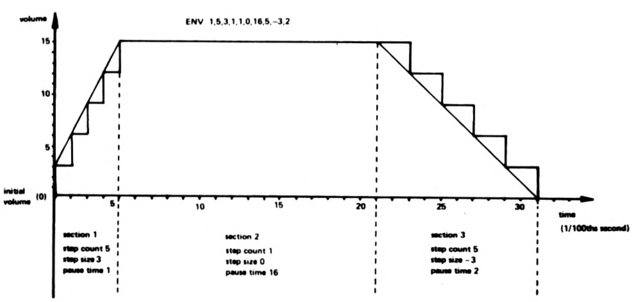
Een integer waarde in het bereik van 0...15.

Standaard waarde, indien niet gegeven: 0.

Bepaalt de ruis die aan SOUND toegevoegd wordt. Als niets of de waarde 0 aangeven wordt, dan wordt er geen ruis aan toegevoegd. Er kan maar altijd een ruistoonhoogte gedefinieerd worden. Dat betekent dat alle kanalen van dezelfde ruis worden voorzien.

6.11 De ENV instructie en geluidssterktevariaties

Aanzwellen en wegsterven van de geluidssterkte geven een toon pas leven. Met deze instructie (ENV= ENvelope Volume) kan men de vorm van een toon bepalen. Tekent u eerst eens de vorm van de toon op papier met gebruikmaking van de waarden voor de diverse instructies. Zie het voorbeeld hieronder:



Beeld 3 : voorbeeld voor de geluidssterktevariatie

De figuur moet in getallen omgezet worden, om ze in een instructie te kunnen verwerken. Verdeelt u de figuur in een aantal delen, tot 5 stuks. In ieder deel heeft de figuur een rechte lijn. Vedeelt u nu ieder deel in stappen waarbij het aantal van de gekozen stappen als <stapgetal> genoemd wordt en de lengte van een stap de <pauze-lengte> is (1=1/100ste seconde).
De stappen nemen met een bepaalde waarde, de <stapgrootte>, toe of af. Als in een deel geen stappen aangegeven zijn wordt de geluidssterkte niet veranderd en aan het volgende deel onveranderd door gegeven.

Let u op: Als men probeert een muziekinstrument te simuleren, dan moet men vaak bij iedere noot aan het begin en het einde de geluidssterkte op nul zetten. Daarom zou dan de begin-geluidssterkte in de SOUND instructie op nul gezet moeten worden en alle verdere geluids-sterkten alleen met de variatie bepaald worden.

Vorm van de instructie

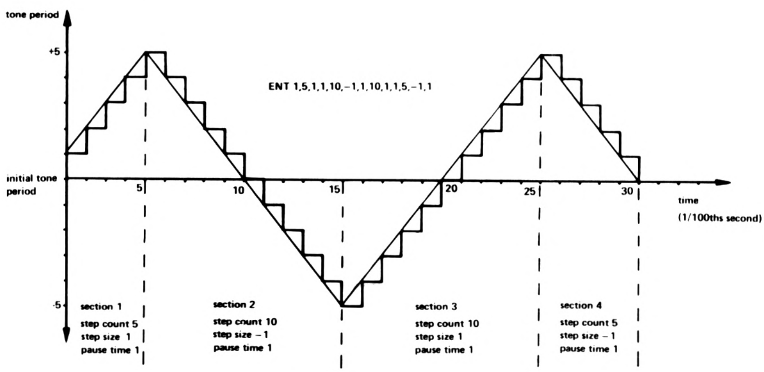
ENV N,P1,Q1,R1,P2,Q2,R2,P3,Q3,R3,P4,Q4,R4,P5,Q5,R5

N	: nummer van de variatie		bereik 1.....15
P1...5:	aantal stappen	(delen 1..5)	bereik 0.....127
Q1...5:	stapgrootte	(delen 1..5)	bereik -128..+127
R1...5:	pauzelengte	(delen 1..5)	bereik 0.....255

Het nummer van de variatie moet worden aangegeven. Evenzo moet een compleet deel aangegeven worden, het aantal van de delen is naar keuze. Als u bijv. de stukken 4 en 5 weg laat, dan zijn er maar 3 stukken.

6.12 De ENT instructie en toonhoogtevariaties

Deze zijn precies zo opgebouwd als geluidsterktevariaties, alleen dat hier kleine toonhoogtevariaties van een noot geproduceerd worden, zoals bij vibrato. Als u voor een toonhoogtevorm van een noot gekozen heeft, schetst u het en verdeelt u het in delen en stappen. Hieronder een voorbeeld:



Beeld 4: Karakteristiek van een toonhoogtevariatie

Het belangrijkste verschil tussen geluidssterktevariatie en toonhoogte-variatie bestaat daarin, dat toonhoogtevariatie niet de duur van een toon beinvloedt, die voorheen in een SOUND instructie of een geluidssterktevariatie gedefinieerd werd. Als de toonhoogtevariatie voor de toon afgelopen is, blijft de laatste toonhoogte voor de rest van de tijd behouden. Zou ze echter langer zijn, dan de tijdsduur van de noot, dan vervallen de overige delen. Om een toonhoogtevariatie tijdens de duur van het geluid te herhalen gebruikt u een negatieve waarde voor de variatie (let er wel op dat de SOUND instructie niet negatief aangeroepen wordt).

Vorm van de instructie:

ENT S,T1,V1,W1,T2,V2,W2,T3,V3,W3,T4,V4,W4,T5,V5,W5

S	**: nummer van de variatie**		**bereik**	**1.....15 (- v.herhaling)**
T1...5:	**aantal stappen**	**(delen 1..5)**	**bereik**	**0.....239**
V1...5:	**stapgrootte**	**(delen 1..5)**	**bereik**	**-128..+127**
W1...5:	**pauzelengte**	**(delen 1..5)**	**bereik**	**0.....255 (1/100ste sec.)**

S moet worden gegeven. Ieder deel, dat wordt ingevoerd moet volledig worden ingevoerd.

Als een <variatie-nummer> gedefinieerd wordt, worden alle vroegere geRESET. Als men een variatie zonder deel aangeeft, worden alle waarden voor deze variatie op nul gezet.

6.13 Verdere bijbehorende functies en instructies

SQ(x)

x is het kanaal-nummer, een bit-significante waarde: 1,2 of 4.

Deze waarden stellen de kanalen A,B en C voor, zoals in de bit-tabellen voor de kanaal-status in het SOUND commando wordt getoond. Daarmee wordt de status van het kanaal gecontroleerd.

De functie geeft als antwoord een integer in het bereik 0 tot 255. De volgende BITS zijn voor gebruik van de informatie van betekenis:

BIT	0...2	Aantal vrije plaatsen in de geteste wachtrij (waarde 0..4)
BIT	3	Rendezvous met kanaal A
BIT	4	Rendezvous met kanaal B
BIT	5	Rendezvous met kanaal C
BIT	6	Stoppunt (HOLD) aan het begin van de wachtrij
BIT	7	Kanaal in bedrijf

BITS 3-6 (Rendezvous en Hold) werden al bij de kanaal-status beschreven. Men had bij deze test nog de mogelijkheid, de ON SQ GOSUB instructie in te schakelen, dat in de volgende paragraaf wordt beschreven.

ON SQ GOSUB

ON SQ(y) GOSUB <regel nummer>

y is het kanaalnummer: 1,2 of 4

Dit is een onderbrekingsinstructie dat nagaat of er een opening in de rij voor het aangegeven kanaal aanwezig is. Hij onderbreekt dan de SOUND subroutine. De SQ en de SOUND instructie heffen beide deze onderbreking op, en de subroutine eindigt heel normaal met RETURN. Alle kanalen hebben dezelfde prioriteit en als een opening in het gegeven kanaal gevonden wordt, wordt een onderbreking geproduceerd en de routine stopt vanzelf. Daarom moet de subroutine de onderbreking opheffen, als ze weer moet worden gebruikt.

RELEASE

RELEASE z

z geeft kanaal nummer(s) aan en is een integer met waarde 1...7

Zoals bij de SOUND instructie beschreven, kunnen de kanalen van stops voorzien worden. Met RELEASE worden deze stops weer opgeheven. De invoer van de kanalen is BIT-significant en geeft de combinatie van de kanalen weer. RELEASE op een kanaal zonder stop heeft geen uitwerking.

BIT 0: kanaal A
BIT 1: kanaal B
BIT 2: kanaal C

7 PRINTERS EN JOYSTICKS

De CPC464 is uitgerust voor het aansluiten van een of twee joysticks en een parallel-printer met centronics interface.

Onderwerpen in dit hoofdstuk:

* JOYSTICKS
* PARALLEL PRINTER
* INTERFACES

De Schneider joystick JY1 kunt u aanschaffen om op de CPC464 computer-spellen met joystick-bediening te spelen. De JY1 wordt aan de 9 polige BUS aan de achterkant van de computer aangesloten, waarboven USER PORT (I/O) staat. Een tweede joystick kan aan de sokkel van de eerste joystick worden aangesloten. Een beschrijving van alle aansluitingen vindt u in aanhangsel V aan het einde van dit handboek.



The JY1 Joystick

7.1 Joysticks

De ingebouwde software in de Schneider CPC464 maakt gebruik van een of twee joysticks mogelijk, die als een deel van het toetsenbord behandeld worden en daarom met INKEY$ en INKEY zoals bij de normale toetsen kunnen worden afgevraagd. Als er maar een vuurknop aan de joystick zit, wordt hij in de CPC464 terminologie als ´FIRE 2´ beschouwd.

Er is een speciale functie om de joystick direkt af te lezen. Dit is JOY(0) voor de eerste en JOY(1) voor de tweede JOYSTICK. De functie geeft een BIT-significante waarde weer, die de status van de joystick op het moment van het laatste toetsenbord test weergeeft. Daar het toetsenbord 50 keer per seconde beproefd wordt, kunt u er van uit gaan, dat u altijd de laatste status van de joystick-knop krijgt.

De waarde in de kolom KEY (toets) zijn de waarden die u moet gebruiken in de INKEY functie. De waarden die u onder SPIEGEL vindt, zijn de overeenkomstige toetsen op het toetsenbord.

1e JOYSTICK	JOY(0)	KEY	2e JOYSTICK	JOY(1)	KEY	SPIEGEL
OP	bit 0	72	OP	bit 0	48	6
NEER	bit 1	73	NEER	bit 1	49	5
LINKS	bit 2	74	LINKS	bit 2	50	R
RECHTS	bit 3	75	RECHTS	bit 3	51	T
VUUR 2	bit 4	76	VUUR 2	bit 4	52	G
VUUR 1	bit 5	77	VUUR 1	bit 5	53	F

Als de tweede joystick wordt gebruikt, kan de CPC464 geen onderscheid maken, tussen de joystick of het toetsenbord. In de praktijk komt het waarschijnlijk nooit tot zo'n situatie. U kunt dus in plaats van een tweede joystick ook het toetsenbord gebruiken.

Als u de Schneider JY1 gebruikt, dan is de tweede hetzelfde als de eerste, en wordt aan de aansluitplug van de eerste bevestigd. Men heeft voor de tweede joystick geen speciale kabel nodig.

Aan de 9-polige aansluitbus, waarboven USER PORT (I/O) staat, kunnen standaard joysticks worden aangesloten, zoals ze ook bij andere computers gebruikt worden. Weliswaar hebben deze geen aansluiting voor een tweede joystick, tenzij men een adapter gebruikt. U moet toch zo'n joystick niet als tweede joystick aan de zijkant in de Schneider JY1 insteken.

Als u software schrijft, moet u aan het begin van een programma de gebruiker de mogelijkheid geven de joystick of de cursortoetsen voor programmabesturing te gebruiken (de [COPY] toets of een andere aan te geven toets zou als vuur knop gebruikt kunnen worden).

7.2 Printer interface

Aan de CPC464 kan een parallel printer met standaard CENTRONICS interface worden aangesloten. De kabel tussen printer en computer is heel eenvoudig een "een op een verbinding". Dat wil zeggen, dat iedere pen van de ene steker met de overeenkomstige van de andere verbonden is. Op de print van de computer bevinden zich 2 pennen minder dan aan de printer-aansluiting, zodat een genormaliseerde printplaat-connector kan worden gebruikt.

De interface beschrijving vindt u in aanhangsel V

De kabel moet dus pen 1 van de computer verbinden met pen 1 van de printer, 2 met 2 enz.. Pen 18 en 36 van de printer worden niet met de computer verbonden. De onderste pennen-rij aan de computer is vanaf 19 opwaarts genummerd en niet vanaf 18, zoals men wellicht verwachten zou. De bovenste rij eindigt bij 17. Daarmee is gegarandeerd dat elke aansluitingsdraad naar de overeenkomstige op de computer gaat.

Het BUSY signaal (pen 11) wordt gebruikt om te testen of de printer klaar is. De computer wacht wanneer de printer OFF LINE is.

U hoeft geen bijzondere instructies te geven, behalve het juiste kanaalnummer op te geven: (stream)#8:

```
LIST #8
```

Dit bewerktstelligt dat het BASIC programma in het geheugen op de aangesloten printer geLIST wordt als het niet om een beveiligd programma gaat.

Binnen een programma kunt u de printer heel eenvoudig besturen:

```
PRINT #8,"Dit wordt naar de printer gestuurd."
```

Veel printers vervolgen automatisch de zin op de nieuwe regel als het einde van de regel bereikt wordt. Kijkt u hier voor in het printer-handboek. Schneider BASIC zet eveneens de zin voort op een nieuwe regel en wel na een lengte die in het WIDTH commando aangegeven is. De standaardwaarde voor de printerbreedte is 132 en kan op een gewenste nieuwe waarde gezet worden bijv. **WIDTH 80**.

Als u de speciale regelbreedte 255 kiest dan laat Schneider BASIC het afbreken van de regel over aan de printer. BASIC heeft een teller voor de printpositie die met de POS-functie gecontroleerd kan worden.

```
IF POS(#8) > 50 THEN GOTO 100
```

De CPC464 produceert aan het einde van een regel, een "nieuwe regel" teken, CHR$(10) en een "wagen-terug-loop"teken CHR$(13) (LF en CR). De printer heeft normaal een instelschakelaar waarmee de juiste vorm van de invoer gekozen kan worden. Als u PRINT wordt direkt duidelijk, welke de juiste stand is.

7.3 Het printen van graphics

Het printerhandboek beschrijft besturingstekens, die normaal gesproken deze vorm hebben:

```
PRINT CHR$(n)
```

Menige printer kan tekens drukken die lijken op de grafische tekens van de Schneider, zoals ze in aanhangsel III te vinden zijn. Maar het is onwaarschijnlijk dat alle teken-nummers overeenstemmen. U moet dan een eigen tabel met de juiste tekens opbouwen, die met uw printer overeenkomen.

Hoewel de printerinterface zo ontworpen is, dat goedkope matrix printers aangesloten kunnen worden, accepteert zij ook DAISY WHEEL printers met de juiste interface en ook PLOTTERS en inktstraalprinters. De sleutel voor deze compatibiliteit is de genormaliseerde parallel interface.

8 Beknopte beschrijving van de Schneider CPC464 BASIC

EEN OPSOMMING VAN ALLE BASIC INSTRUCTIES MET VOORBEELDEN EN VERWIJZINGEN NAAR AANVERWANTE INSTRUCTIES:

Onderwerpen in dit hoofdstuk:

* BESCHRIJVING VAN DE INSTRUCTIE VORM
* BIJZONDERE TEKENS MET HUN BETEKENIS
* ALLE SCHNEIDER BASIC INSTRUCTIES OP ALFABETISCHE VOLGORDE

Dit hoofdstuk bevat een beknopte samenvatting van alle BASIC instructies meegeleverd in ROM met de CPC464.
De beschikbare instructies omvatten niet alleen standaard BASIC statements maar ook instructies die de bijzondere HARDWARE mogelijkheden van de CPC464 ondersteunen.

8.1 Notatie: verklaring van de bij de beschrijving van de instructies gebruikte tekens

Bijzondere Tekens:

& of &H	kenmerkt navolgende constante als HEXadecimaal
&X	kenmerkt navolgende constante als BINair
:	scheidt de instructies die in dezelfde regel voorkomen
#	kenmerkt de navolgende constante als In-en Uitvoer eenheid (stream)

Data types

Strings kunnen van 0 tot 255 tekens lang zijn en worden uitgebeeld als: <string uitdrukking> (<string expression>). Strings kunnen met een ´+´ teken aan elkaar gekoppeld worden. Daarbij mag de complete lengte de 255 tekens niet overschrijden.

Een <numerieke uitdrukking> (<numeric expression>) kan een natuurlijk getal (real) of geheel getal (integer) zijn, dat wil zeggen de getalwaarden worden respectievelijk met of zonder decimale cijfers uitgebeeld. Een <integer uitdrukking> kan de waarde van -32768 tot +32767 aannemen. Een <real uitdrukking> kan in bereik van -1.7E+38 tot +1.7E+38 liggen en hebben een nauwkeurigheid van iets meer dan negen plaatsen, met het kleinste getal ongelijk nul van ongeveeris 2.9E-39.

% als het laatste teken van een variabele definieert de variabele als een integer. ! als een real (natuurlijk getal) en $ als string-variabele.

Een <numerieke uitdrukking> heeft altijd een getalswaarde als uitkomst. Daarbij kan het een echte getalswaarde bevatten of een variabele (integer of real) of kan het een wiskundige uitdrukking zijn. Kortgezegd alles behalve <stringuitdrukking> is toegestaan.

De <in en uitvoer eenheid> stelt een <numerieke uitdrukking> voor, bijv. het beeldscherm, de printer of het cassettedeck dus het identiteits bewijs van een apparaat.

Een ontoelaatbare uitdrukking ('Improper Argument') betekent dat een <numerieke uitdrukking> een waarde heeft die buiten het toegelaten getalsbereik valt of dat een instructieparameter onjuist ingegeven is.

INSTRUCTIES

SVP let u er op dat alle Schneider BASIC instructies als volgt zijn opgebouwd.

Instructies
Instructie schrijfwijze / functie
Voorbeeld
Beschrijving
Aanverwante instructies:

Belangrijk:

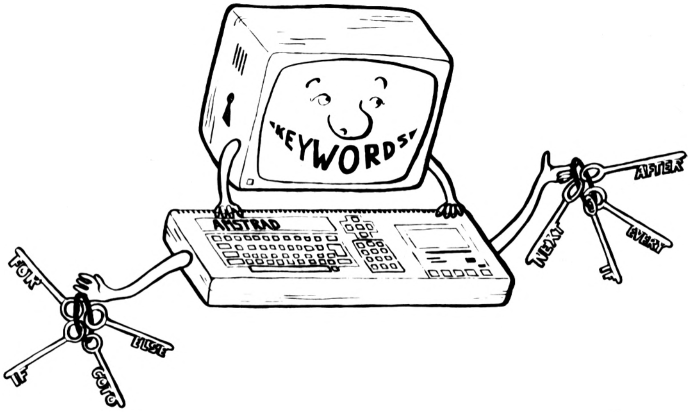
Instructies zijn of
Opdrachten: die direkt kunnen worden uitgevoerd of
Functies: die een waarde bevatten in een uitdrukking

Haakjes
() zijn delen van een instructie of een functie. Andere soorten haakjes binnen een instructie notatie hebben alleen een betekenis als beschrijving en mogen daarom niet worden ingevoerd.
[] Omsluiten te kiezen uitdrukkingen.
<> Omsluiten verschillende uitdrukkingen die in de instructie beschrijving verklaard worden.

Aanhalings tekens
Alleen " " (dubbele aanhalingstekens) zijn onderdeel van BASIC.
"(enkelvoudig aanhalingsteken) dient alleen in bijzondere gevallen binnen de beschrijving doch mogen nergens in een instructie voorkomen het zij dan binnen een <string uitdrukking>.

Invoer
BASIC zet alle in kleine letters ingegeven instructies om in hoofdletters, als een programma geLIST wordt. De gebruikte voorbeelden zijn alle in hoofdletters, zoals ze bij de LISTing verschijnen. Indien u de voorbeelden in kleine letters invoert, is het mogelijk fouten veel sneller te vinden, omdat deze foutief ingegeven instructies bij een LISTing in klein letters verschijnen.

Instructies moeten door spaties gevolgd worden, omdat het bij Schneider BASIC geoorloofd is BASICwoorden in variabelen te gebruiken:
Bijv. **end2** en **listcode** worden door Schneider BASIC als variabelen aangenomen.



ABS

ABS(<numerieke uitdrukking>)

PRINT ABS (-67.98)
67.98

FUNCTIE: geeft de absolute waarde van de numerieke uitdrukking terug, namelijk van Negatief naar Positief.

Aanverwante instructie: **SGN**

AFTER

AFTER <integer uitdrukking>[,<integer uitdrukking>] GOSUB<regelnummer>

AFTER 200,2 GOSUB 320

OPDRACHT: Roept na een bepaalde tijd een subroutine op. De eerste <integer uitdrukking> beschrijft de tijdsinterval in eenheden van 0.02 seconden. De tweede <integer uitdrukking> (bereik van 0...3) geeft de timer (4 mogelijkheden) aan die gebruikt kan worden.

Aanverwante instructies: **EVERY,REMAIN**

ASC

ASC (<string uitdrukking>)

PRINT ASC("X")
88

FUNCTIE: Geeft een numerieke waarde van het eerste teken van de string terug, indien ASCII tekens gebruikt worden.

Aanverwante instructie: **CHR$**

ATN

ATN (<numerieke uitdrukking>)

PRINT ATN(1)
0.785398163

FUNCTIE: Berekent de arctangens van de ingegeven waarde, waarbij een natuurlijk getal in radialen wordt teruggegeven met een waarde tussen -PI/2 en +PI/2.

Aanverwante instructies: **SIN,COS,TAN,DEG,RAD**

AUTO

AUTO [<regelnummer>][,<ophoging>]

AUTO 100,50

OPDRACHT: Geeft AUTOmatisch de regelnummers. <regelnummer> geeft het eerste regelnummer aan, waarmee begonnen wordt, ingeval u bijv. wilt toevoegen aan het eind van een programma. <ophoging> geeft waarde van ophoging tussen regelnummers. Worden beiden of een der beiden niet ingevoerd dan wordt 10 als standaardwaarde genomen. Indien een regel met hetzelfde regelnummer bestaat zal BASIC dit aangeven met een * teken na het regelnummer, als waarschuwing voor overschrijving.

BIN$

BIN$ (<integer uitdrukking zonder teken> [,<integer uitdrukking>])

PRINT BIN$(64,8)
01000000

FUNCTIE: geeft een string binaire tekens die de waarde vertegenwoordigen van de <integer uitdrukking zonder teken>, waarbij het getal eventueel wordt vooraf gegaan door nullen om het geheel op te vullen tot het aantal cijfers dat de tweede <integer uitdrukking> aangeeft, inclusief de geldende voorafgaande nullen.

Aanverwante instructies: **HEX$,STR$**

BORDER

BORDER <kleur>[,<kleur>]

BORDER 3,2

OPDRACHT: Veranderd de kleur van de buitenrand(border) van het beeldscherm in de aangegeven <kleur>. Wordt een tweede <kleur> toegevoegd dan wisselt de buitenrand altijd tussen de beide kleuren. De tijdsinterval tussen de kleurenwissel wordt door de instructie SPEED INK vastgelegd. <kleur> mag de waarde 0...26 hebben.

Aanverwante instructie: **SPEED INK**

CALL

CALL <adres>[,lijst van:<parameter>]

CALL &BD19

OPDRACHT: Springt naar de machinetaalroutine buiten het BASIC-programma dat op het aangegeven adres start. Bij gebruik van deze instructie is voorzichtigheid geboden en zal alleen door ervaren programmeurs gebruikt worden. De als voorbeeld gebruikte CALL is relatief gezien ongevaarlijk omdat het wacht op de volgende beeldopbouw (pas als een elektronenstraal het nieuwe beeld helemaal heeft opgebouwd wordt het zichtbaar). Het is daarmee mogelijk om tekens op het beeldscherm zo natuurgetrouw mogelijk te laten bewegen.

Aanverwante instructie: **UNT**

CAT

CAT

CAT

OPDRACHT: Leest alle gevonden namen van programma´s die op cassette staan. De instructie heeft geen uitwerking op het programma dat reeds in het geheugen zit. Daarmee kan de instructie gebruikt worden om te controleren of een geSAVEd programma ook goed is opgeslagen op de cassette, voordat men het geheugen voor een ander programma gebruikt. De instructie verlangt dat de PLAY toets van de cassette ingedrukt wordt, dan een willekeurige toets ("Press PLAY then any key") waarna ieder gevonden blok vermeld wordt in de vorm:
Filenaam   Blok nummer   vlag   OK

De vlaggen geven de soort van de opgeslagen gegevens aan:
$ BASIC programma
% beveiligd BASIC programma
*** ASCII tekst file**
& binaire file
Andere tekens kunnen voorkomen, wanneer het een file betreft die niet onder BASIC is ontstaan.

Aanverwante instructies: **LOAD,RUN,SAVE**

CHAIN
CHAIN MERGE

CHAIN <filenaam>[,<regelnummer uitdrukking>]
CHAIN MERGE <filenaam>[,<regelnummer uitdrukking>][,DELETE <regel-nummergebied>]

```
 CHAIN "TEST", 350
```

OPDRACHT: Chain laadt een programma <filenaam> van de cassette in het geheugen, waarbij het bestaande programma gewist wordt. CHAIN MERGE voegt het programma <filenaam> van de cassette samen met het bestaande programma. De <regelnummer uitdrukking> geeft aan met welke regel het programma na het laden beginnen zal. Onbreekt de startregel dan begint BASIC bij het laagste regelnummer. Met DELETE <regelnummergebied> wordt voor het laden het aangegeven gebied gewist. Wordt geen <filenaam> aangegeven dan laadt BASIC het eerste programma in dat hij tegen komt. Indien voor de <filenaam> een ! gegeven wordt dan worden de verdere meldingen van het cassette lezen onderdrukt.
CHAIN MERGE houdt alle tot nu toe gebruikte variabelen in stand, maar de zelf gedefinieerde functies en de open filestreams zullen verloren gaan. Een geactiveerde ON ERROR GOTO-sprong wordt uitgeschakeld, een RESTORE wordt uitgevoerd en de met DEFINT, DEFREAL en DEFSTR geactiveerde variabelentype-definitie worden geRESET. Bovendien worden alle actieve FOR-,WHILE- en GOSUB- instructies vergeten. Beveiligde BASIC programma´s kunnen niet geMERGEd worden.

Aanverwante instructies: **LOAD,MERGE**

CHR$

CHR$(<numerieke uitdrukking>)

```
 PRINT CHR$(100)
 d
```

FUNCTIE: Geeft het character terug uit de Schneider CPC464 charactertabel van de <integer uitdrukking> (zie ook aanhangsel III).

Aanverwante instructies: **ASC,LEFT$,RIGHT$,MID$,STR$**

CINT

CINT (<numerieke uitdrukking>)

```
 10 n=578.76543
 20 PRINT CINT(n)
 RUN
  579
```

FUNCTIE: geeft de afgeronde integer waarde van de <numerieke uitdrukking> in het bereik van -32768 tot +32767

Aanverwante instructies: **CREAL,INT,FIX,ROUND,UNT**

CLEAR

CLEAR

CLEAR

OPDRACHT: Wist alle variabelen en files.

CLG

CLG <pen>

CLG

OPDRACHT: Vult het grafisch scherm met de opgegeven <pen> (INK). Is geen <pen> opgegeven, dan wordt de bij de vorige CLG-instructie opgegeven <pen> gebruikt, anders <pen> 0

Aanverwante instructies: **CLS,ORIGIN**

CLOSEIN

CLOSEIN

CLOSEIN

OPDRACHT: Sluit de data-invoer vanuit de cassetterekorder af. Instructies zoals NEW en CHAIN MERGE vergeten iedere geopende file.

Aanverwante instructies: **OPENIN,CLOSEOUT**

CLOSEOUT

CLOSEOUT

CLOSEOUT

OPDRACHT: sluit de data-uitvoer naar de cassetterecorder af.

Aanverwante instructies: **OPENOUT,CLOSEIN**

CLS

CLS [#<in- uitvoereenheid>]

CLS

OPDRACHT: wist de met <in- uitvoereenheid> aangegeven schermwindow met de voor de achtergrond vastgelegde kleur (paper ink).

CONT

CONT

CONT

OPDRACHT: zet een door *BREAK*, STOP, of END afgebroken programma voort, indien in het programma geen veranderingen zijn aangebracht. Direkte instructies mogen tussen BREAK en CONT worden gebruikt.

COS

COS (<numerieke uitdrukking>)

```
?COS(34)
 -0.848570274

deg:?cos(34)
 0.829037573
```

FUNCTIE: geeft de waarde van de cosinus <numerieke uitdrukking>. De functie werkt met radialen tenzij anders, als aangegeven met DEG. Let u er op dat de in het voorbeeld gebruikte ? de verkorte vorm is van PRINT en het gebruik van kleine letters voor de instructies, een mogelijkheid is die Schneider BASIC volledig begrijpt.

Aanverwante instructies: **SIN,TAN,ATN,DEG,RAD**

CREAL

CREAL (<numerieke uitdrukking>)

```
 5 DEFINT n
 10 n=75.765
 20 d=n/34.6
 30 PRINT d
 40 PRINT CREAL(n)
 50 PRINT n/55.4
 run
  2.19653179
  76
  1.37184116
 Ready
```

FUNCTIE: zet de <numerieke uitdrukking> om in een natuurlijk getal (het tegengestelde van integer). Het getal krijgt weer een waarde achter de komma.

Aanverwante instructie: **CINT**

DATA

DATA <lijst van:<constanten>>

```
 10 REM namenlijst
 20 DIM VOORNAAM$(3)
 30 DIM ACHTERNAAM$(3)
 40 FOR n=1 TO 3
 50 READ VOORNAAM$(n)
 60 READ ACHTERNAAM$(n)
 65 PRINT VOORNAAM$(n);" ";ACHTERNAAM$(n)
 70 DATA Bert,Smit,Dick,Jansen,Willem,Molendijk
 90 NEXT
```

OPDRACHT: maakt de erachter staande gegevens tot afleesbare informatie in het programma. Het is een veel gebruikte mogelijkheid in BASIC. Data achter de DATA instructie kunt u op elk moment in een programma aanroepen. De gegeven DATA moet met de aard van de variabelen overeenkomen, die u wilt gebruiken. DATA-instructies kunnen overal in het programma voorkomen.

Aanverwante instructies: **READ,RESTORE**

DEF FN

DEF FN <naam>[(<formule parameters>)]=<algemene uitdrukking>

```
 10 DEF FNrente(bedrag)=0.14*bedrag
 20 INPUT "Hoe hoog is het bedrag";bedrag
 30 PRINT "Het totaal aan rente na een jaar is";FNrente(bedrag)
```

OPDRACHT: BASIC maakt het mogelijk in het programma functies te definieren die een waarde terug geven. De met DEF FN-instructies bepaalde functies gelden binnen het programma en werken op de zelfde wijze als de in de BASIC ingebouwde functies (bijv. COS). De functies kunnen overal in het programma gebruikt worden. De te gebruiken variabelen moeten in soort met de in de functie gebruikte waarden overeenstemmen.

DEFINT
DEFSTR
DEFREAL

DEFtype <lettergebied(en)>

```
 DEFINT I-N
 DERSTR A,W-Z
 DEFREAL
```

OPDRACHT:legt de soort van alle variabelen vast die met een aangegeven <letter> beginnen van de variabelen <type>, waarbij INT integer, STR string (tekst) en REAL natuurlijk getal (met decimale komma (REAL)) betekent. Daarbij wordt geen onderscheid gemaakt tussen de grote en kleine letters.

Aanverwante instructies: **LOAD,RUN,CHAIN,NEW,CLEAR**

DEG

DEG

DEG

OPDRACHT: schakelt om naar graden. Standaard wordt met radialen gerekend in functies als SIN en COS. De instructie schakelt om naar graden, totdat door CLEAR, RAD of het laden van een nieuw programma weer wordt teruggeschakeld naar RADialen.

Aanverwante instructie: **RAD**

DELETE

DELETE <regelnummergebied>

DELETE 100-200

OPDRACHT: verwijdert alle regels die worden aangegeven in het <regelnummergebied>. Daar die regels nu verwijderd zijn moet men ze weer opnieuw invoeren indien men ze achteraf toch nodig heeft.
Wees er dus voorzichtig mee !

Aanverwante instructie: **NEW**

DI

DI

```
10 CLS
20 TAG
30 EVERY 10 GOSUB 100
40 X1=RND*320:X2=RND*320
50 Y=200+RND*200
60 FOR X=320-X1 TO 320+X2 STEP 2
70 DI:PLOT 320,0:MOVE X-2,Y:PRINT " ";:MOVE X,Y:PRINT "#";:EI
80 NEXT
90 GOTO 40
100 MOVE 320,0
110 DRAW X+8,Y-16,0
120 RETURN
```

OPDRACHT: Schakelt alle onderbrekingen (interrupts) uit behalve die van ***BREAK*** totdat het door EI of de RETURN instructie van een interrupt-GOSUBroutine weer ingeschakeld wordt. De instructie wordt gebruikt om een programma zonder onderbreking af te laten werken, indien bijv. programmadelen een en het zelfde onderdeel willen gebruiken . In het voorbeeld gebruiken het hoofdprogramma en de subroutine beide het grafische beeldscherm.

Aanverwand bevel: **EI**

DIM

DIM <lijst van:<geindexeerde variabelen>

```
 10 CLS:PRINT "Geef vijf namen...":PRINT
 20 DIM B$(5)
 30 FOR N=1 TO 5
 40 PRINT "Naam ";N;"a.u.b";
 50 INPUT B$(N)
 60 NEXT
 70 FOR N=1 TO 5
 80 PRINT "Heel goed van u ";B$(N);" om de CPC464 te kopen"
 90 NEXT
```

OPDRACHT: Het statement DIM wordt gebruikt voor het vaststellen van het aantal elementen dat is toegestaan in een numerieke of een string-array. Een array dimensionering geschiedt door plaatsing van de array-variabele achter het DIM statement, gevolgd door de arraygrootte tussen haakjes. Bijv. DIM a(20) staat variabele a toe om 21 array-elementen te gebruiken van a(0) tot a(20). Wordt geen arrayveld aangegeven dan neemt de BASIC dit op als DIM a(10). Wanneer het DIM statement wordt uitgevoerd zet de computer de waarden die zijn opgeslagen in ieder benoemd array-element op nul. Array is de in de computerwereld gebruikte naam voor matrix.

Aanverwante instructie: **ERASE**

DRAW

DRAW <x-coordinaat>,<y-coordinaat>[,<pen>]

DRAW 200,200,13

OPDRACHT: tekent een lijn vanaf de cursorpositie tot aan de aangegeven positie (x,y) op het grafische beeldscherm. De coordinaten zijn voor alle drie de beeldschermen hetzelfde. Verdere voorbeelden vindt u in Hoofdstuk 5.

Aanverwante instructies: **DRAWR,PLOT,PLOTR,MOVE,MOVER,TEST,TESTR,
XPOS,YPOS,ORIGIN**

DRAWR

DRAWR <x-verplaatsing>,<y-verplaatsing>[,<pen>]

DRAWR 200,200,13

OPDRACHT: tekent een lijn vanaf de cursorpositie tot de relatieve positie, aangegeven door <x-verplaatsing> en <y-verplaatsing>.

Aanverwante instructies: **DRAW,PLOT,PLOTR,MOVE,MOVER,TEST,TESTR,
XPOS,YPOS,ORIGIN**

EDIT

EDIT <regelnummer>

EDIT 110

OPDRACHT: het wijzigen van een specifieke programmaregel. Zie ook Hoofdstuk 1

Aanverwante instructie: **LIST**

EI

EI

EI

OPDRACHT: om onderbrekingen na de DI instructie weer mogelijk te maken.

Aanverwante instructie: **DI**

END

END

END

OPDRACHT: stopt het programma. Een programma stopt ook na de laatste regel. END sluit alle files en schakelt terug naar de direkte mode. Nog aanwezige toonreeksen worden tot aan het einde uitgevoerd.

Aanverwante instructie: **STOP**

ENT

ENT <variatie-nummer>[,<variatie-volgorde>]

```
10 ENT 1,100,2,20
20 SOUND 1,100,1000,4,0,1
```

OPDRACHT: maakt het mogelijk een gegeven toon in hoogte te varieren. Een toon <variatie-volgorde> geeft aan hoe de toonhoogte gevarieerd wordt. Een uitvoerige beschrijving hiervan vindt u in Hoofdstuk 6 en aanhangsel 7. Het <variatie-nummer> is een <integer uitdrukking>, die een absolute waarde tussen 1 en 15 moet hebben en kenmerkt de toonvariatie-volgorde. Indien het <variatie-nummer> negatief is wordt de <variatie-volgorde> herhaald.

Tot 5 <variatie-volgorden> kunnen telkens aangegeven worden in de vorm van <aantal stappen>,<stapgrootte>,<pauzelengte> of:= <toonhoogte>, <pauzelengte>.
De eerste vorm bepaalt een stapsgewijze verandering in een momentele toonhoogte, de tweede bepaalt de absolute toonhoogte. Daarbij betekent het <aantal stappen> een <integer uitdrukking> die tussen 0 en 239 ligt en die het aantal uitvoerende eenheden voorstelt. <stapgrootte> is een <integer uitdrukking> die tussen -128 en +127 ligt en het verschil tussen de beide toonhoogten voorstelt.

De <pauzelengte> is een <integer uitdrukking> tussen 0 en 255 en die de wachttijd tussen de beide toonveranderingen voorstelt. Daarbij wordt de tijd in eenheden van 0.01 seconden gegeven. Bovendien wordt de 0 als 256 gerekend.

<Toonhoogte> is een <integer uitdrukking> die tussen 0 en 4095 ligt en de nieuwe toonhoogte aangeeft.

De SOUND instructie bepaalt de toonhoogte bij de start en kan een van de 15 toonhoogtevariaties bepalen. Wordt geen of een niet gedefinieerde variatie aangegeven, dan behoudt de gegeven toon altijd zijn originele waarde (toonhoogte).

Een toonhoogtevariatie heeft geen uitwerking op de duur van de toon zelf. Zijn er nog stappen in de veranderingsvolgorde over, terwijl de toon reeds gestopt is dan worden deze niet meer uitgevoerd.

Een herhalende toonhoogtevariatie wordt steeds weer van voren af aan gestart, zolang de toon duurt.

De in de instructie aangegeven waarden worden bij de uitvoering bewaard en kunnen later altijd weer aangeroepen worden. De instructie hoeft dan niet worden herhaald.

Iedere keer als er een toonhoogtevariatie met een gelijk <variatienummer> gedefinieerd wordt, worden de oude waarden overschreven. Vindt zo´n variatie plaats tijdens een SOUND instructie dan onstaat er een in principe onmogelijke combinatie met mogelijk een interessant effekt.

Wordt een ENT instructie zonder <variatie-volgorde> aangegeven dan worden de bewaarde waarden van het aangesproken <volgnummer> gewist.

Aanverwante instructies: **ENV,SOUND**

ENV

ENV <variatienummer>[,<variatievolgorde>]

```
 10 ENV 1,100,2,20
 20 SOUND 1,100,1000,4,1
```

OPDRACHT: maakt het mogelijk een gegeven toon in geluidssterkte te varieren en geeft aan, hoe de geluidssterkte veranderd moet worden. Daarbij kan het aantal stappen liggen tussen 0 en 127. De <stapgrootte> ligt dan tussen -128 en +127 en de pauzetijd in eenheden van 0.01 sekonden tussen 1 en 256.

Het <variatienummer> is een <integer uitdrukking>, waarvan de waarde tussen 1 en 15 moet liggen en bepaalt de geluidssterktevariatie.

Tot en met 5 <variaties> kunnen telkens aangegeven worden in de vorm:
<aantal stappen>,<stapgrootte>,<pauzelengte>

of:= **<registerwaarde>,<variatieperiode>**.

De eerste vorm geeft een variatie via de software. Daarbij betekent:
<aantal stappen> een <integer uitdrukking> die ligt tussen 0 en 127 en die het aantal stappen bepaalt.
<stapgrootte> een <integer uitdrukking> die tussen -128 en +127 ligt en die het verschil tussen de verschillende stappen van de geluidsvariatie voorstelt. In het geval <aantal stappen> gelijk aan 0 is geeft het de absolute instelling aan.

<Pauzelengte> een <integer uitdrukking> die tussen 0 en 255 ligt en die de wachttijd tussen de stappen bepaalt. Daarbij wordt de tijd in eenheden van 0.01 seconden gerekend. Bovendien wordt 0 als 256 gerekend.

De tweede vorm bepaalt een variatieperiode door direkte ingreep in de toongenerator. Daarbij bepaalt <registerwaarde> de in het geluidssterkteregister <register 15 octaal> te plaatsen waarde.
<variatieperiode> de in de variatievolgorderegister (registers 13 en 14 octaal) te plaatsen waarde.

Doet men dit direct in het hardwareregister, dan wordt de geluidssterktevariatie meteen uitgevoerd. Het is daarom raadzaam, voor de volgende geluidssterktevariatie een pauze in te lassen. Zijn er geen verdere geluidssterktevariaties dan wordt een pauze van 2 sekonden aangehouden.

De SOUND instructie bepaalt de geluidssterktevariatie en kan een der 15 variatienummers aanspreken. Wordt er geen of een niet gedefinieerd variatienummer gegeven, dan blijft de geluidssterkte altijd hetzelfde Wordt er een <stapgrootte> met waarde 0 en een <aantal stappen> ongelijk 0 aangegeven, dan blijft de geluidssterkte ook gelijk.

De in de instructie aangegeven waarden worden bij de uitvoering opgeslagen en kunnen later altijd weer worden opgeroepen. De instructie hoeft dan niet meer herhaald te worden.

Iedere keer als er een geluidsvariatievolgorde met een gelijk <variatienummer> gedefinieerd wordt zal de oude waarde overschreven worden. Vindt zo'n verandering plaats indien de SOUND instructie in gebruik is dan zal er een bepaald effekt onstaan.
Wordt een ENV instructie zonder <variatienummer> aangegeven dan zal de opgeslagen waarde altijd het eerder bepaalde <variatienummer> wissen.

Aanverwante instructies: **ENT,SOUND**

EOF

EOF

```
 PRINT EOF
  -1
```

FUNCTIE:geeft de waarde -1 (TRUE) als bij invoer van files vanaf cassette het einde bereikt is, anders is de waarde 0 (FALSE).

Aanverwante instructie: **OPENIN**

ERASE

ERASE <lijst van <variabele namen>>

```
 ERASE A,B$
```

OPDRACHT: wist de aangegeven variabelen. Indien een array niet meer nodig is kan het gewist worden om het geheugen weer voor iets anders te gebruiken.

Aanverwante instructie: **DIM**

ERR
ERL

ERR
ERL

```
10 CLS
20 ON ERROR GOTO 1000
30 DATA SALLY,EMMA,JOANNA,HELEN,GEMMA
40 READ A$
50 PRINT A$
60 GOTO 40
70 REM hier begint de error behandeling
1000 IF ERR=4 THEN PRINT "Ik heb een fout ontdekt in een dataregel!"
1005 PRINT "Ik kom gegevens te kort"
1010 IF ERL<70 AND ERL>20 THEN PRINT "De fout zit in de regels 30-60"
1020 END
```

VARIABELEN: deze variabelen worden in het foutmeldingssysteem gebruikt om vast te stellen in welke regel, welke fout met welk foutnummer zit. Foutmeldingen worden in aanhangsel VIII beschreven.

Aanverwante instructies: **ON ERROR,ERROR**

ERROR

ERROR <integer uitdrukking>

ERROR 17

OPDRACHT: voert de behandeling van een fout met dit nummer uit. De foutmelding kan er een zijn die al gebruikt is en door BASIC herkent wordt (zie ook aanhangsel VIII). In dit geval wordt de daarvoor bepaalde behandeling uitgevoerd. In andere gevallen wordt de fout alleen gegeven en het programma gestopt.

Aanverwante instructies: **ON ERROR,ERR,ERL**

EVERY

EVERY <integer uitdrukking>[,<integer uitdrukking>] GOSUB<regelnummer>

EVERY 500,2 GOSUB 50

OPDRACHT: de CPC464 heeft een ingebouwde timer. De EVERY instructie maakt het mogelijk in een BASIC programma op regelmatige tijden een subroutine op te roepen. Er zijn 4 verschillende timers ter beschikking, die door de waarde van 0 tot 3 met de tweede <integer uitdrukking> aan de subroutine worden gekoppeld.

Aanverwante instructies: **AFTER,REMAIN**

EXP

EXP <numerieke uitdrukking>

```
PRINT EXP(6.876)
 968.743625
```

FUNCTIE: de functie EXP berekent de macht van het getal E, het grondgetal van de natuurlijke logaritme. Deze functie is tegengesteld aan de LOG functie.

Aanverwante instructie: **LOG**

FIX

FIX <numerieke uitdrukking>

```
PRINT FIX(9.99999)
 9
```

FUNCTIE: wordt gebruikt om alle cijfers achter de decimale punt te verwijderen. De FIX instructie geeft dus integer waarden terug, zonder af te ronden als bij CINT.

Aanverwante instructies: **CINT,INT,ROUND**

FOR

FOR <enkelvoudige variabele> = <start> TO <einde> [STEP <grootte>]

FOR dag=1 to 5 STEP 2

OPDRACHT: herhaalt een programmadeel zo vaak als door de <start> en <einde> waarden aangegeven zijn. Indien geen <grootte> wordt aangegeven, heeft deze de waarde 1.

Aanverwante instructies: **NEXT,WHILE**

FRE

FRE(<numerieke uitdrukking>)

FRE(<string uitdrukking>)

PRINT FRE(0)
PRINT FRE("")

FUNCTIE: geeft het aantal nog vrije geheugenplaatsen aan in BASIC. FRE("") (garbage collection) verzamelt alle lege geheugenplaatsen en voegt deze aan het totaal toe alvorens de totale vrije ruimte wordt afgedrukt.

GOSUB

GOSUB <regelnummer>

GOSUB 210

OPDRACHT: roept een subroutine aan die op het gegeven <regelnummer> begint. (Zie ook RETURN)

Aanverwante instructie: **RETURN**

GOTO

GOTO <regelnummer>

GOTO 90

OPDRACHT: springt direkt naar het aangegeven <regelnummer>.

HEX$

HEX$(<integer uitdrukking zonder voorteken>[,<integer uitdrukking>])

```
PRINT HEX$(65534)
 FFFE
```

FUNCTIE: geeft de waarde van <integer uitdrukking zonder voorteken> in HEXadecimale vorm terug (zie ook aanhangsel II). De tweede <integer uitdrukking> bepaalt het aantal van de terug te geven tekens en dient tussen 0 en 16 te liggen of weggelaten te worden.

Aanverwante instructies: **BIN$,STR$**

HIMEM

HIMEM

```
PRINT HIMEM
 43903
```

VARIABELE: Geeft het hoogste geheugenadres van het door BASIC gebruikte geheugen en kan in numerieke uitdrukkingen op de gebruikelijke manier gebruikt worden.
Het is aan te bevelen, voordat de hoogste byte van de beschikbare geheugenruimte met MEMORY wordt veranderd, het statement mm=HIMEM te gebruiken om later met MEMORY mm de oorspronkelijke geheugenruimte terug te krijgen.

Aanverwante instructies: **FRE,MEMORY**

IF

IF

IF <logische uitdrukking> THEN <instructie> [ELSE<instructie>]

IF <logische uitdrukking> GOTO <regelnummer> [ELSE <instructie>]

```
IF A>B THEN A=C ELSE A=D
IF A>B GOTO 1000 ELSE 300
```

OPDRACHT: afhankelijk van de test van de <logische uitdrukking> wordt de ene of de andere instructie uitgevoerd. Is de <logische uitdrukking> waar dan wordt het deel na THEN of GOTO uitgevoerd. In het andere geval wordt het gedeelte na ELSE uitgevoerd of, indien er geen ELSE is, dan wordt er verder gegaan met de volgende regel. Een IF instructie is niet langer dan de regel en mag geen verdere instructies bevatten die niet samenhangen met IF...THEN.

Aanverwante instructies: **THEN,ELSE,GOTO,OR,AND,WHILE**

INK

INK <pen>,<kleur>[,<kleur>]

```
INK 0,1
```

OPDRACHT: afhankelijk van de momentele beeldschermmode kunnen een aantal INKs beschikbaar zijn (zie hoofdstuk 5). De kleur, of kleuren, die voor een <pen> gebruikt worden kunnen door de INK instructie worden gewijzigd, overeenkomstig de kleurentabel in aanhangsel IV.

Aanverwante instructies: **PEN,PAPER**

INKEY

INKEY (<integer uitdrukking>)

```
10 CLS:IF INKEY(55)=32 THEN 30 ELSE 20
20 GOTO 10
30 PRINT "U hebt [SHIFT] en V gedrukt"
40 FOR t=1 TO 1000:NEXT:GOTO 10
```

FUNCTIE:Het toetsenbord wordt iedere 0.02 seconde gecontroleerd en geeft dan aan welke toets er op het toetsenbord is ingedrukt. De functie kan worden gebruikt om ja-of nee-antwoorden te testen, waarmee het niet zo belangrijk is of er kleine-of hoofd-letters ingevoerd worden. Het bovengenoemde voorbeeld controleert of [SHIFT] en V gelijktijdig worden ingedrukt. De teruggegeven waarden van de toetsen zijn:

Waarde teruggegeven	[SHIFT]	[CTRL]	TOETS
-1	?	?	niet ingedrukt
0	niet ingedrukt	niet ingedrukt	ingedrukt
32	ingedrukt	niet ingedrukt	ingedrukt
128	niet ingedrukt	ingedrukt	ingedrukt
160	ingedrukt	ingedrukt	ingedrukt

Aanverwante instructies: **INPUT,INKEY$**

INKEY$

INKEY$

```
10 CLS
20 PRINT "Bent u slim (j/n)?"
30 a$=INKEY$:IF a$="" GOTO 30
40 IF a$="N" OR a$="n" THEN PRINT "U moet het geweest zijn,
   want u hebt mij gekocht !":END
50 IF a$="J" OR a$="j" THEN PRINT "U bent wel bescheiden !!!":END
60 GOTO 20
```

FUNCTIE: leest een karakter van het toetsenbord zonder dat de [ENTER] toets na ieder antwoord moet worden ingedrukt. Als een toets is ingedrukt, dan geeft het de waarde van de invoer anders geeft het een lege string. Door deze lege string blijft de computer in een lus hangen totdat een geldige invoer van het toetsenbord is ontvangen.

Aanverwante instructies: **INPUT,INKEY**

INP

INP(<poort adres>)

```
PRINT INP(&FF77)
```

FUNCTIE: Geeft de invoerwaarde terug van de I/O poort aangegeven door <poort adres>.

Aanverwante instructies: **OUT,WAIT**

INPUT

INPUT[#<stream>,][;][<te printen tekst>;]<lijst van[variabelen]>
INPUT[#<stream>,][;][<te printen tekst>,]<lijst van[variabelen]>

```
10 CLS
20 INPUT "Geef twee door een komma gescheiden getallen";A,B
30 IF A=B THEN PRINT "De beide getallen zijn gelijk"
40 IF A>B THEN PRINT A "is groter dan" B
50 IF A<B THEN PRINT A "is kleiner dan" B
60 CLEAR:GOTO 20
```

OPDRACHT: Leest data in van de opgegeven ´stream´. Een puntkomma na INPUT onderdrukt de "wagenterug/nieuwe regel" (carriage return) aan het einde van de in te voeren regel. Een puntkomma na de <te printen tekst> laat een vraagteken afdrukken, een komma houdt dit tegen. Als een foutieve invoer is gegeven (bijv. een letter O i.p.v. 0 (nul) in een <numerieke uitdrukking>, dan laat BASIC dit weten door:

? Redo from start

en de vraag die u aan het beantwoorden was.
Alle antwoorden moeten afgesloten worden met [ENTER]. De puntkomma direct na <stream> onderdrukt de carriage return, de cursor blijft achter de afgedrukte tekst staan. Indien een cassette´stream´ is aangegeven, dan wordt <te printen tekst> niet afgedrukt. Zo kan het zelfde programma iedere ´stream´ uitlezen.

Per bestanddeel van de ingelezen file wordt een variabele uit de <lijst van[variabelen]> genomen. Het moet overeenstemmen met het variabeletype gespecificeert in de INPUT instructie, welke is: een nummerieke variabele, afgesloten door of komma, [ENTER], spatie of een End Of File. Komma´s of [ENTER] na een spatie worden buiten beschouwing gelaten. Tekst die niet in aanhalingstekens staat wordt met komma of spatie opgedeeld in enkele tekens (als bij numerieke waarden).

Aanverwante instructies: **LINE INPUT,READ,INKEY$**

INSTR

INSTR([<integer uitdrukking>,]<stringuitdrukking>,<stringuitdrukking>)

```
PRINT INSTR(2,"BANANA","AN")
```

FUNCTIE: Geeft het positienummer terug, waar de tweede <string uitdrukking> gevonden werd. Het zoeken begint bij de eventueel door de <integer uitdrukking> aangegeven positie, anders bij het begin van de eerste <string uitdrukking> .

Aanverwante instructies: **MID$,LEFT$,RIGHT$**

INT

INT(<numerieke uitdrukking>)

```
PRINT INT(-1.995)
 -2
```

FUNCTIE: Rondt af tot op de eerstvolgende kleinere integer waarde. INT werkt hetzelfde als de FIX instructie alleen dat voor negatieve getallen een afronding naar een kleinere waarde gegeven wordt, indien ze niet integer zijn.

Aanverwante instructies: **CINT,FIX,ROUND**

JOY

JOY(<integer uitdrukking>)

```
10 IF JOY(0)=8 THEN GOTO 100
```

FUNCTIE: leest het bitsignificante resultaat van de joystick (<integer waarde> = 0 of 1).

BIT	Betekenis	Decimale waarde
0	omhoog	1
1	omlaag	2
2	links	4
3	rechts	8
4	vuren 2	16
5	vuren 1	32

Aanverwante instructie: **INKEY**

KEY

KEY <integer>,[CHR$(n)+]<string uitdrukking>[+CHR$(n)]

```
KEY 140,"RUN"+CHR$(13)
```

OPDRACHT: maakt van een toets een functietoets. Het geeft de mogelijkheid om 32 functietoetsen te maken in het ASCII bereik van 128 tot 159, hetgeen beschreven is in aanhangsel III. De som van de opgeslagen uitbreidingstekens is voorlopig beperkt tot 120 tekens. Door de functietoets in te drukken wordt de eronder opgeslagen tekst geactiveerd.

Aanverwante instructie: **KEY DEF**

KEY DEF

KEY DEF <toetsnummer>,<herhaling>[,<normaal teken>[,<met shift>[,<met CTRL>]]]

```
KEY DEF 46,1,63
```

OPDRACHT: verandert de aan een toets toegekende waarde (zie aanhangsel III). Het voorbeeld hierboven laat zien hoe bij het indrukken van toets N het ? (teken 63) wordt gegeven. Om deze toets weer haar oorspronkelijk functie te geven schrijft men:

```
KEY DEF 46,1,110.
```

Waarbij het teken N 110 betekend.

Aanverwante instructie: **KEY**

LEFT$

LEFT$(<string uitdrukking>,<integer uitdrukking>)

```
10 CLS
20 A$ = "SCHNEIDER"
30 B$ = LEFT$(A$,3)
40 PRINT B$
RUN
      [wis beeldscherm]
 SCH
 Ready
```

FUNCTIE: geeft uit de <string uitdrukking> van het begin af zoveel tekens terug als door <integer uitdrukking> wordt aangegeven. Als de <string uitdrukking> korter is dan de gewenste lengte dan wordt de gehele <string uitdrukking> terug gegeven.

Aanverwante instructies: **MID$,RIGHT$**

LEN

LEN(<string uitdrukking>)

A$="SCHNEIDER":PRINT A$

FUNCTIE: geeft het aantal van alle tekens, inclusief spaties, die de <string uitdrukking> bevat.

LET

LET <logische uitdrukking>

LET X=100

OPDRACHT: verouderde vorm uit de eerste BASICs om variabelen een waarde toe te kennen. De instructie is opgenomen om compatibel te zijn met vroegere BASIC-trainingsboeken. Het voorbeeld moet bij de Schneider ingevoert worden als:

x=100

LINE INPUT

LINE INPUT[<#<stream>,][;][<te printen tekst>;]<stringvariabele>
LINE INPUT[<#<stream>,][;][<te printen tekst>,]<stringvariabele>

LINE INPUT A$
LINE INPUT "NAAM"; N$

OPDRACHT: leest een regel van de aangegeven <stream>. Een LINE INPUT onderdrukt de "wagenterug/nieuwe regel"(CR/LF) na de invoer. Is geen <stream> aangegeven dan wordt standaard #0 , het beeldscherm, genomen.

Aanverwante instructies: **READ,INPUT,INKEY$,INPUT$**

LIST

LIST [< regelbereik>][,#<stream>]

LIST 100-1000,#1

OPDRACHT: brengt de programmaregels op de vooraf bepaalde ´stream´. #0 is het beeldscherm, #8 is de printer. De LISTing kan door eenmaal op de toets [ESC] te drukken onderbroken worden en door op de spatie-balk te drukken weer verder gaan. Tweemaal drukken op de [ESC] toets laat het LISTen afbreken. Programma´s kunnen in van te voren gedefineerde beeldschermbereiken (WINDOW) uitgeLIST worden, om bijv. fouten te zoeken, zonder het gehele beeldscherm te overschrijven. U kunt de volgende instructies eventueel invoeren:

LIST,LIST -200,LIST 30,LIST 30-200,LIST 200-

LOAD

LOAD <filenaam>[,<adres uitdrukking>]

```
 LOAD "PROGRAMMA 1"
```

OPDRACHT: leest een BASIC programma van cassette in het geheugen en wist daarbij een bestaand programma of, indien er een <adres uitdrukking> aangegeven is wordt er een binaire file in het geheugen geladen. (Zie ook Hoofdstuk 2).

LOCATE

LOCATE [#<stream>,]<x-coordinaat>,<y-coordinaat>

```
 10 MODE 1
 20 LOCATE 20,12
 30 PRINT CHR$(249)
```

OPDRACHT: zet de cursor op de aangegeven plaats van de x-en y-coordinaten, die t.o.v. het ORIGIN punt in het beeldschermbereik (WINDOW) wordt bepaald. De linker bovenhoek van de WINDOW heeft de coordinaten 1,1. Is er geen <stream> aangegeven, dan wordt standaard #0 aangenomen.

Aanverwante instructie: **WINDOW**

LOG

LOG (<numerieke uitdrukking>)

```
 ?LOG(9999)
  9.21024037
```

FUNCTIE: geeft de natuurlijke logaritme van de <numerieke uitdrukking> als natuurlijk getal.

Aanverwante instructies: **EXP,LOG 10**

LOG10

LOG10(<numerieke uitdrukking>)

```
?LOG10(9999)
 3.99995657
```

FUNCTIE: geeft de logarithme met basis 10 van de <numerieke uitdrukking> als natuurlijk getal.

Aanverwante instructies: **READ,LOG**

LOWER$

LOWER$(<string uitdrukking>)

```
A$="SCHNEIDER":PRINT LOWER$(A$)
 schneider
```

FUNCTIE: geeft de gegeven <string uitdrukking> van hoofdletters of gemengde schrijfwijze terug in kleine letters. Een nuttig hulpmiddel om invoer te testen, die op verschillende wijze zijn ingevoerd.

Aanverwante instructie: **UPPER$**

MAX

MAX (<lijst van:<numerieke uitdrukkingen>)

```
10 n=66
20 PRINT MAX(1,n,3,6,4,3)
```

FUNCTIE: geeft de grootste waarde uit een lijst van <numerieke uitdrukkingen) terug.

Aanverwante instructie: **MIN**

MEMORY

MEMORY <adres uitdrukking>

```
MEMORY &20AA
```

OPDRACHT: stelt de bovengrens van het BASIC geheugen vast (zie ook HIMEM). Om het geheugenbereik te controleren gebruikt men de FRE instructie.

Aanverwante instructies: **HIMEM,FRE**

MERGE

MERGE [<filenaam>]

MERGE "PLAN"

OPDRACHT: koppelt een programma van cassette aan een programma dat reeds in het geheugen zit. De inhoud van de file wordt dus bij het bestaande programma toegevoegd. Indien er geen <filenaam> wordt gegeven wordt het eerstvolgende programma van de cassette geladen. Mocht er voor de filenaam een ! teken staan dan wordt deze van de filenaam verwijderd en worden de scherm-meldingen over het laden van de cassette onderdrukt. Indien u een programma aan een ander in het geheugen wilt toegevoegen zonder het te overschrijven kunt u het beste eerst het huidige programma met RENUM van andere regelnummers regelnummers voorzien in een regelnummerbereik boven dat van het te MERGEn programma. Alle variabelen, gebruikersfuncties en open files worden gewist. ON ERROR GOTO wordt uitgeschakeld, een RESTORE uitgevoerd en de DEFINT,DEFREAL & DEFSTR instructies worden geRESET. Beveiligde files (programma's) kunnen niet met de MERGE instructie ingelezen worden.

Aanverwante instructies: **LOAD,CHAIN,CHAIN MERGE**

MID$

MID$ (<string>,<integer uitdrukking>[,<integer uitdrukking>])

```
A$="SCHNEIDER":PRINT MID$(A$,2,4)
 CHNE
A$="SCHNEIDER":b$=MID$(A$,2,4):PRINT b$
 CHNE
```

FUNCTIE: geeft vanaf de door de eerste <integer uitdrukking> aangegeven positie van de <string> een aantal tekens terug. De tweede <integer uitdrukking> bepaalt het aantal tekens die worden gegeven (standaard: tot aan het einde van de <string>).

Aanverwante instructies: LEFT$,RIGHT$

MIN

MIN (<lijst van <numerieke uitdrukkingen>>)

? MIN(3,6,2.999,8,9)

FUNCTIE: geeft de kleinste waarde van de <numerieke uitdrukkingen> terug.

Aanverwante instructie: **MAX**

MOD

MOD

MOD

MODE

MODE<integer uitdrukking>

MODE 1

OPDRACHT: verandert de beeldscherm modus (0,1 of 2) en wist het beeldscherm met INK 0. Alle WINDOWs worden op beeldschermgrootte en de cursor op de home positie gezet.

Aanverwante instructies: **WINDOW,ORIGIN**

MOVE

MOVE <x-coordinaat>,<y-coordinaat>

MOVE 34,34

OPDRACHT: zet de graphics-cursor op de positie van de aangegeven absolute x,y coordinaat.

Aanverwante instructies: **MOVER,PLOT,PLOTR,DRAW,DRAWR,ORIGIN,TEST, TESTR,XPOS,YPOS**

MOVER

MOVER <x-verplaatsing>,<y-verplaatsing>

MOVER 34,34

OPDRACHT: zet de graphics-cursor van zijn huidige positie naar een relatieve x,y coordinaat, die zich uit de x,y verplaatsing laat berekenen.

Aanverwante instructies: **MOVE,PLOT,PLOTR,DRAW,DRAWR,ORIGIN,TEST, TESTR,XPOS,YPOS**

NEW

NEW

NEW

OPDRACHT: wist het in het geheugen zijnde programma en tevens alle variabelen. KEY definities blijven behouden. Het beeldscherm wordt niet gewist. Wij raden u aan om het met uiterste voorzichtigheid te gebruiken.

NEXT

NEXT [<lijst van:<variabelen>>]

```
FOR n=1 TO 1000:NEXT
```

OPDRACHT: bepaalt het einde van de FOR...NEXT lus. De NEXT instructie kan aangegeven worden met of zonder variabelenaam, die verwijst naar de bijbehorende FOR instructie. In bovenstaand voorbeeld zou NEXT dan de volgende vorm hebben:

```
NEXT n
```

Aanverwante instructie: **FOR**

ON GOSUB
ON GOTO

ON <integer uitdrukking> GOSUB <lijst van:<regelnummers>>
ON <integer uitdrukking> GOTO <lijst van:<regelnummers>>

```
10 ON dag  GOSUB 100,200,300,400,500,1000
10 ON bereik GOTO  150,250,350,450,550,1050
```

OPDRACHT: springt afhankelijk van <integer uitdrukking> naar een subroutine (GOSUB) of naar een regel in het programma (GOTO). Indien de waarde van de <integer uitdrukking> 1 is, dan wordt er naar het eerste regelnummer gesprongen, bij 2 naar de tweede enz.. In bovenstaande regel 10 zal wanneer dag=1 de subroutine op regel 100 aangeroepen worden, dag=2 de subroutine op regel 200 enz..

Aanverwante instructies: **GOTO,GOSUB**

ON BREAK GOSUB

ON BREAK GOSUB <regelnummer>

```
10 ON BREAK GOSUB 40
20 PRINT "Het programma werkt"
30 GOTO 20
40 CLS
50 PRINT "Tweemaal [ESC] drukken roept de subroutine op"
60 FOR t=1 TO 2000:NEXT
70 RETURN
```

OPDRACHT: springt naar een subroutine op het <regelnummer> wanneer er tweemaal op de [ESC] toets gedrukt wordt.

Aanverwante instructies: **ON BREAK STOP,RETURN**

ON BREAK STOP

ON BREAK STOP

```
10 ON BREAK GOSUB 40
20 PRINT "Het programma werkt"
30 GOTO 20
40 CLS
50 PRINT "Tweemaal [ESC] drukken roept de subroutine op"
60 FOR t=1 TO 2000:NEXT
65 ON BREAK STOP
70 RETURN
```

OPDRACHT: stopt het programma als er tweemaal op de toets [ESC] gedrukt wordt. In het voorbeeld springt het programma bij de eerste dubbele [ESC] naar de subroutine, en bij de tweede maal wordt het programma gestopt. In het bovenstaand voorbeeld werkt ON BREAK GOSUB dus slechts een keer

Aanverwante instructie: **ON BREAK GOSUB**

ON ERROR GOTO

ON ERROR GOTO <regelnummer>

```
10 ON ERROR GOTO 80
20 CLS
30 PRINT "Indien ik een fout gemaakt heb"
40 PRINT "wil ik het programma uitlisten, om"
50 PRINT "te zien waar de fout zit."
60 FOR t=1 TO 4000:NEXT
70 GOTO 200
80 CLS:PRINT "Fout in regel";ERL:PRINT
90 LIST
```

OPDRACHT: springt bij het optreden van een fout naar de aangegeven regel. In het voorbeeld hierboven zit de fout in regel 70.

Aanverwante instructies: **ERR,ERL,RESUME**

ON SQ GOSUB

ON SQ (<kanaal>) GOSUB <regelnummer>

```
ON SQ (2) GOSUB 2000
```

OPDRACHT: springt naar de subroutine op de aangegeven regel, indien er in de SOUND wachtrij van het aangegeven <kanaal> een onderbreking voorkomt. Het <kanaal> is een <integer uitdrukking> met de volgende waarden:

1	voor Kanaal	A
2	voor Kanaal	B
4	voor Kanaal	C

Aanverwante instructies: **SOUND,SQ**

OPENIN

OPENIN <filenaam>

100 OPENIN "!INFORMATIE"

OPDRACHT: opent een invoerfile van cassette. Indien voor <filenaam> een ! wordt gezet, dan worden de overige meldingen van het cassette lezen onderdrukt en het programma leest het eerste 'block' met de <filenaam> van de cassette.

Aanverwante instructies: **CLOSEIN,OPENOUT**

OPENOUT

OPENOUT <filenaam>

100 OPENOUT "!DATA1"

OPDRACHT: opent een uitvoerfile op de cassette. Indien voor de <filenaam> een ! gezet wordt, dan worden de overige meldingen onderdrukt en het programma schrijft het eerste 'block' met de <filenaam> op de cassette weg. Ieder 'block' is 2 K groot. Een 'block' wordt pas geschreven, als de 2K buffer vol is, of als de file met de CLOSEOUT instructie wordt gesloten.

Aanverwante instructies: **CLOSEOUT,OPENIN**

ORIGIN

ORIGIN <x>,<y>[,<links>,<rechts>,<boven>,<onder>]

```
 10 CLS:BORDER 13
 20 ORIGIN 0,0,50,590,350,50
 30 DRAW 540,350
 40 GOTO 20
```

OPDRACHT: bepaalt het startpunt van de graphics-cursor. Het [keuze-deel] bevat de instructiemogelijkheden voor het bepalen van een nieuwe graphics-WINDOW, die in alle modes blijft bestaan dankzij de gebruikte beeldpunt-adresseer-techniek.

Het nulpunt (ORIGIN) is een punt met de coordinaten 0,0 in de linkeronderhoek van de WINDOW, de coordinaten gaan naar rechts en naar boven. Liggen de hoeken buiten het beeldschermbereik dus buiten het echte beeldscherm dan worden ze nog geaccepteerd. De tekeningen zijn daar echter niet zichtbaar.

Aanverwante instructie: **WINDOW**

OUT

OUT <poort nummer>,<integer uitdrukking>

```
 OUT &F8F4,10
```

OPDRACHT: stuurt de waarde van de <integer uitdrukking>, die tussen 0 en 255 moet liggen, naar de I/O poort die wordt gespecificeerd in <poort nummer> met het adres.

Aanverwante instructies: INP,WAIT

PAPER

PAPER [#<stream>,]<PEN>

```
 10 MODE 0
 20 FOR p=0 TO 15
 30 PAPER p:CLS
 40 PEN 15-p
 50 LOCATE 6,12:PRINT "PAPER"p
 60 FOR t=1 TO 500:NEXT t
 70 NEXT p
```

OPDRACHT: Wordt er een teken op het beeldscherm geschreven dan wordt de bij het teken behorende achtergrond met de <PEN> (INK) bepaald. Behalve wanneer er in de transparant mode wordt gewerkt.

Het te gebruiken aantal van <PEN> hangt af van de gekozen beeldscherm-modus. Indien de aangegeven <PEN> niet beschikbaar is, wordt er automatisch een andere kleur gekozen.

Aanverwante instructies: **INK,WINDOW,PEN**

PEEK

PEEK (<adres uitdrukking>)

```
 10 MODE 2
 20 INK 1,0: INK 0,12: BORDER 12
 30 INPUT "Startadres van het geheugenbereik"; begin
 40 INPUT "Eindadres van het geheugenbereik"; einde
 50 FOR n= begin TO einde
 60 VALUE$=HEX$(PEEK(n),2)
 70 PRINT VALUE$;
 80 PRINT " op ";HEX$(n,4),
 90 NEXT
```

FUNCTIE: geeft de inhoud van een aangegeven geheugenadres weer. Met het voorbeeld kan men de inhoud van het RAM-geheugen van de CPC464 bekijken. Daarmee is het mogelijk het RAM-bereik te lezen, onder de onderste (van &0000 tot &3FFF) en de bovenste (van &C000 tot &FFFF). ROM, niet de ROM zelf.

Aanverwante instructie: **POKE**

PEN

PEN [#<stream>,]<INK>

```
 PEN 1,2
```

OPDRACHT: bepaalt de kleur die gebruikt wordt bij het schrijven van tekens naar de gegeven WINDOW. Is geen <stream> aangegeven dan wordt standaard #0 aangenomen (beeldscherm).

Aanverwante instructies: **INK,PAPER**

PI

PI

```
 PRINT PI
  3.14159265

 10 REM perspectief tekening
 20 MODE 2
 30 RAD
 40 INK 1,0
 50 INK 0,12
 60 BORDER 0
 70 FOR N=1 TO 200
 80 ORIGIN 420,0
 90 DRAW 0,200
 100 REM teken hoeken van verdwijnpunt
 110 DRAW 30*N*SIN(N*PI/4),(SIN(PI/2))*N*SIN(N)
 120 NEXT
 130 MOVE 0,200
 140 DRAWR 0,50
 150 DRAWR 40,0
 160 WINDOW 1,40,1,10
 170 PRINT "Schrijft u het programma Galgje nu zelf verder !"
```

FUNCTIE: geeft de waarde van PI, de verhouding tussen omvang en doorsnede van een cirkel: 3.141592653468251. De functie wordt in de meeste grafische programma´s gebruikt.

Aanverwante instructies: **DEG,RAD**

PLOT

PLOT <x-coordinaat>,<y-coordinaat>[,<PEN>]

```
10 MODE 2:PRINT "Geef 4 getallen gescheiden door komma's":PRINT
20 PRINT "Invoer van x-middel punt (0-639), y-middel punt (0-399),
   straal en stapgrootte":INPUT x,y,r,s
30 ORIGIN x,y
40 FOR hoek =1 TO 360 STEP s
50 XPUNT = r*COS(hoek)
60 YPUNT = r*SIN(hoek)
70 PLOT XPUNT,YPUNT
74 REM MOVE 0,0
75 REM DRAW XPUNT,YPUNT
80 NEXT
```

OPDRACHT: probeert u 320,200,20,1 als eerste antwoord. PLOT doet hetzelfde als DRAW, behalve dat alleen het beeldpunt op de aangegeven x,y coordinaat wordt getekend. Verwijdert u in regel 75 REM om de DRAW instructie uit te voeren. U moet dan wel in regel 70 REM voor de PLOT instructie invoeren. U kunt nu het verschil duidelijk zien (als u de REM instructie ook in regel 74 verwijderd, wordt uw cirkel opgevuld). Let er wel op dat het programma de cirkel in meerdere lussen uitvoert, omdat RAD niet is uitgeschakeld, zodat ´hoek´ bij iedere stap meer dan 1 graad is. Voegt u nu in regel 25 de instructie DEG eens toe en laat het programma dan nog eens lopen.

Aanverwante instructies: **DRAW,DRAWR,PLOTR,MOVE,MOVER,ORIGIN,TEST, TESTR,XPOS,YPOS**

PLOTR

PLOTR <x-opvulling>,<y-opvulling>[,<PEN>]

```
10 MODE 2:PRINT "Geef 4 getallen gescheiden door komma's":PRINT
20 PRINT "Invoer van x-middel punt (0-639), y-middel punt (0-399),
    straal en stapgrootte":INPUT x,y,r,s
30 ORIGIN x,y
40 FOR hoek =1 TO 360 STEP s
50 XPUNT = r*COS(hoek)
60 YPUNT = r*SIN(hoek)
70 PLOTR XPUNT,YPUNT
80 NEXT:GOTO 40
```

OPDRACHT: plot ieder punt van de momentele cursorpositie tot aan de aangegeven relatieve x,y coordinaten. De PLOTR instructie is gelijk aan de DRAWR instructie, behalve dat alleen het beeldpunt op de x,y coordinaat wordt getekend. Probeert u eens in het voorbeeld de volgende waarden 320,0,20,1

Aanverwante instructies: **DRAW,DRAWR,PLOT,MOVE,MOVER,ORIGIN,TEST, TESTR,XPOS,YPOS**

POKE

POKE <adres uitdrukking>,<integer uitdrukking>

POKE &00FF,10

OPDRACHT: schrijft de <integer uitdrukking>, waarvan de waarde tussen 0 en 255 moet liggen, in de aangegeven geheugenplaats. Bij het gebruiken van deze instructie moet men voorzichtig te werk gaan, omdat bij het schrijven naar bepaalde adressen het systeem geblokkeerd kan raken.

Aanverwante instructie: **PEEK**

POS

POS (#<stream>)

PRINT POS(#0)
1

FUNCTIE: geeft de huidige positie van de aangegeven <stream> terug. De <stream> moet altijd aangegeven worden, anders wordt er een **Syntax error** gemeld.

Het beeldscherm: geeft de huidige X coordinaat van de tekstcursor in relatie tot de huidige WINDOW. Daarbij is origin altijd de linkerbovenhoek van de WINDOW: 1,1.

De printer: geeft de huidige positie van de schrijfkop, waarbij 1 de eerste plaats aangeeft in de regel. Alle tekens groter dan &1F(31) worden geteld.

De cassette: als voor de printer.

Aanverwante instructie: **VPOS**

PRINT

PRINT [#<stream>,][<print lijst>][<USING<formaat>][<scheidingsteken>]

PRINT #0,"abc"

OPDRACHT: print naar een aangegeven <stream>. Een uitvoerige beschrijving vindt u op bladzijde 54 van dit hoofdstuk.

Aanverwante instructies: **USING,TAB,SPC**

RAD

RAD

```
RAD
```

OPDRACHT: schakelt de hoekmodus om naar radialen.

Aanverwante instructies: DEG,SIN,COS,TAN,ATN

RANDOMIZE

RANDOMIZE [<numerieke uitdrukking>]

```
10 RANDOMIZE 23
20 PRINT RND(6)
```

OPDRACHT: zet de startwaarde van de willekeurige-getallen-generator. In BASIC worden willekeurige getallen afhankelijk van een aangegeven startwaarde gegeven. De volgorde is altijd hetzelfde. Met de RANDOMIZE instructie wordt een nieuwe startwaarde genomen of wel door een gegeven waarde of een in te voeren getal. RANDOMIZE TIME produceert een serie van willekeurige getallen, die moeilijk te herhalen zijn.

Aanverwante instructie: RND

READ

READ <lijst van <variabelen>>

```
10 FOR X=1 TO 4
20 READ N$
30 PRINT N$
40 DATA Bert,Henk,Klaas,Ria
50 NEXT
```

OPDRACHT: leest gegevens van een DATA regel en zet deze in de gegeven <lijst van <variabelen>>. De RESTORE instructie zet de pointer voor de te lezen gegevens weer op het eerste gegeven van de daarmee aangegeven regel (zie ook de DATA instructie).

Aanverwante instructies: DATA,RESTORE

RELEASE

RELEASE <kanalen>

RELEASE 4

OPDRACHT: heft de eventuele wachttijd op van de aangegeven geluids <kanalen>. De waarden om de <kanalen> aan te spreken zijn Bit-afhankelijk. Namelijk:

```
Kanaal A= bit 0   (1)
Kanaal B= bit 1   (2)
Kanaal C= bit 2   (4)
```

Zo geeft 4 (binair 0100) kanaal C aan.

Aanverwante instructie: **SOUND**

REM

REM <rest van de regel>

```
10 REM De hier gegeven tekst dient als
20 REM programmadocumentatie
```

OPDRACHT: voegt commentaarregels toe aan het programma. Het teken (') in een regel heeft de betekenis: REM tenzij dat het is geplaatst in een <string> of een DATA regel.

REMAIN

REMAIN (<integer uitdrukking>)

```
REMAIN (3)
PRINT #6,REMAIN(0);
```

FUNCTIE: geeft de overgebleven wachttijd van de door de <integer uitdrukking> (tussen 0 en 3) bepaalde timer en schakelt hem uit. Indien de aangesproken timer niet ingeschakeld was wordt 0 gegeven.

Aanverwante instructies: **AFTER,EVERY**

RENUM

RENUM [<nieuw regelnummer>][,[<oud regelnummer>][,<stapgrootte>]]

```
RENUM
RENUM 100,,100
```

OPDRACHT: hernummert het programma. De <nieuwe regelnummer> geeft het eerste regelnummer aan. De standaardwaarde is 10. Met het <oude regelnummer> kan aangegeven worden, bij welke regels begonnen moet worden. De <stapgrootte> bepaalt de afstand tussen de regelnummers. RENUM corrigeert automatisch alle GOSUB en GOTO instructies. Indien geen parameter aangegeven wordt, dan wordt het programma als RENUM 10,,10 hernummerd. De regelnummers kunnen in het bereik van 1 tot 65535 liggen.

RESTORE

RESTORE [<regelnummer>]

```
RESTORE 300

10 FOR N=1 TO 6
20 READ A$
30 PRINT A$ "  ";
40 DATA restored, gegevens, kan, men, weer, lezen
50 NEXT
60 PRINT
70 RESTORE
80 GOTO 10
```

OPDRACHT: zet de pointer voor de te lezen gegevens op de eerste in het programma optredende of op de met het <regelnummer> aangegeven DATA.

Aanverwante instructies: **READ,DATA**

RESUME

RESUME [<regelnummer>]
of RESUME NEXT

```
RESUME 300
```

OPDRACHT: Is een fout ontdekt met ON ERROR GOTO dan zorgt RESUME er voor dat het programma verder kan gaan, eventueel vanaf een op te geven <regelnummer>.

Aanverwante instructie: **ON ERROR GOTO**

RETURN

RETURN

```
RETURN
```

OPDRACHT: geeft het eind aan van een subroutine en keert uit de subroutine terug om voorbij de instructie GOSUB weer verder te gaan.

Aanverwante instructies: **GOSUB,ON X GOSUB,ON SQ GOSUB, AFTER n GOSUB, EVERY n GOSUB,ON BREAK GOSUB**

RIGHT$

RIGHT$ (<string>,<integer uitdrukking>)

```
10 CLS
20 A$ = "SCHNEIDER"
30 B$ = RIGHT$(A$,3)
40 PRINT B$
RUN
         [wis scherm]
DER
Ready
```

FUNCTIE: geeft het met <integer uitdrukking> aangegeven aantal van tekens, vanuit het einde van de <string> gerekend, terug. Indien de <string> korter is dan het gevraagde aantal tekens, dan wordt de gehele <string> gegeven.

Aanverwante instructies: **MID$,LEFT$**

RND

RND [(<numerieke uitdrukking>)]

```
10 RANDOMIZE 23
20 PRINT RND(6)
```

FUNCTIE: geeft het eerstvolgende willekeurige getal terug. Kan een volgende uit de reeks, een herhaling van de laatste of een eerste van een nieuwe reeks zijn. De RANDOMIZE instructie in het bewuste voorbeeld zorgt er voor, dat RND(6) altijd dezelfde waarde terug geeft, namelijk 0.146940658.
RND(0) geeft het vorige willekeurige getal nog eens. Is de <numerieke uitdrukking> negatief dan is de reeks gegenereerde willekeurige getallen voorspelbaar.

Aanverwante instructie: **RANDOMIZE**

ROUND

ROUND (<numerieke uitdrukking>[,<integer uitdrukking>])

```
10 x=0.123456789
20 FOR r=9 TO 0 STEP -1:PRINT r,ROUND(x,r):NEXT
25 x=132456789
30 FOR r=0 TO -9 STEP -1
40 PRINT r,ROUND (x,r)
50 NEXT
```

FUNCTIE: geeft de afgeronde waarde van een <numerieke uitdrukking>. Daarbij wordt afgerond op het aantal cijfers, door <integer uitdrukking> bepaald. Indien de <integer uitdrukking> kleiner dan 0 is dan wordt de <numerieke uitdrukking> tot een integer waarde afgerond, waarbij het gegeven aantal plaatsen voor de decimale punt door een nul opgevuld wordt tot het aantal tekens gelijk aan de <integer uitdrukking> is. De plaatsen achter de komma vervallen.

Aanverwante instructies: **INT,FIX,CINT,ABS**

RUN "

RUN <string uitdrukking>

```
RUN "PROGRAMMA 1"
```

OPDRACHT: laadt een programma van de cassette en start het. Is de <string uitdrukking> leeg ("") dan laadt BASIC het eerstgevonden programma. Wordt voor de programmanaam een ! gegeven, dan worden de verdere meldingen van het cassettelezen onderdrukt.

OPMERKING: BASIC wist alle in het geheugen zijnde programma's bij het laden van cassette.

Aanverwante instructie: **LOAD**

RUN

RUN [<regelnummer>]

```
RUN 100
```

OPDRACHT: start het in het geheugen zijnde programma van het begin of vanaf het aangegeven <regelnummer>. Alle programma- definities, functies en variabelen in het geheugen worden gewist. DEFINT, DEFREAL en DEFSTR definities verliezen hun geldigheid. Geopende files worden afgebroken evenals de cassettebuffers.

Aanverwante instructie: **LOAD**

SAVE

SAVE <filenaam>[,<filetype>][,<binaire parameters>]

```
 SAVE "PROGRAMMA",P
```

OPDRACHT: schrijft het in het geheugen aanwezige programma op de cassette weg in een file met de aangegeven <filenaam>. Als filetype kan daarbij opgegeven worden:

- **,A** voor SAVEn als ASCII-file
- **,P** voor SAVEn als beveiligd programma
- **,B** voor het opslaan als binaire file (bijv. het beeldscherm)

Daarbij beschrijven de <binaire parameters> het startadres en de lengte in bytes van de te SAVEn file.

Aanverwante instructies: **LOAD,RUN"<filenaam>",MERGE,CHAIN,CHAIN MERGE**

SGN

SGN (<numerieke uitdrukking>)

```
 10 INPUT "Hoe is het huidige banksaldo";CASH
 20 IF SGN(CASH) <1 GOTO 30 ELSE 40
 30 PRINT "Oh, oh":END
 40 PRINT "U hebt meer geld dan ik"
```

FUNCTIE: geeft -1,indien de <numerieke uitdrukking> kleiner is dan 0 en 0 indien de <numerieke uitdrukking> gelijk is aan 0. Geeft +1 indien de <numerieke uitdrukking> groter is dan 0.

Aanverwante instructie: **ABS**

SIN

SIN <numerieke uitdrukking>

```
 PRINT SIN(PI/2)
  1
```

FUNCTIE: geeft de sinus van de <numerieke uitdrukking> in radialen, tenzij DEG is bepaald.

Aanverwante instructies: **COS,TAN,ATN,DEG,RAD**

SOUND

SOUND <kanaalstatus>,<toonhoogte>[,<lengte>[,<geluidssterkte>[,<geluidssterktevariatie>[,<toonhoogtevariatie>[,<ruistoonhoogte>]]]]]

```
SOUND 1,200,1000,7,0,0,1
```

OPDRACHT: uitvoering van geluid. De muziekmogelijkheden van de CPC464 zijn een van de meest complexe uitbreidingen van BASIC en zijn beschreven in hoofdstuk 6.

Aanverwante instructies: **ENV,ENT**

SPACE$

SPACE$ (<integer uitdrukking>)

```
SPACE$(190)
```

FUNCTIE: geeft een <string uitdrukking> terug, gevuld met de door de <integer uitdrukking> aangegeven aantal spaties.

Aanverwante instructies: **PRINT,SPC,TAB**

SPEED INK

SPEED INK <integer uitdrukking>,<integer uitdrukking>

```
5 INK 0,9,12:INK 1,0,26
10 BORDER 12,9
20 SPEED INK 50,20
```

OPDRACHT: stelt de snelheid van het kleurwisselen vast voor de INK en BORDER instructies. de INK en BORDER instructies kunnen beide wisselen in kleuren. De eerste <integer uitdrukking> geeft de tijdsduur van de eerste kleur aan, de tweede <integer uitdrukking> de tijdsduur van de tweede kleur. Er wordt gemeten in eenheden van 0.02 seconde. Gaat u voorzichtig om met het uitzoeken van de kleurwisselingen, daar u hele rare effekten kunt krijgen.

Aanverwante instructies: **INK,BORDER**

SPEED KEY

SPEED KEY <startvertraging>,<herhalingssnelheid>

SPEED KEY 20,3

OPDRACHT: bepaalt de herhalingssnelhied van de ingedrukt gehouden toets en de tijdvertraging voordat de herhaling in werking treedt. De tijden worden daarbij in eenheden van 0.02 sekonden gemeten. De aangegeven waarden moeten daarbij tussen 1 en 255 liggen. De standaard waarden zijn altijd 10.

Een zeer kleine <startvertraging> kan het onmogelijk maken het toetsenbord te bedienen daar de software niet de hardware kan beinvloeden, om een bepaalde toets te kunnen lezen. Niet alle toetsen zijn voor herhaling bedoeld. De KEY DEF instructie zorgt er voor dat de gebruiker de herhaling van een toets in en uit kan schakelen.

Aanverwante instructie: **KEY DEF**

SPEED WRITE

SPEED WRITE <integer uitdrukking>

SPEED WRITE 1

OPDRACHT: bepaalt de schrijfsnelheid voor het SAVEn op cassette. Er kan met een snelheid van 2000 baud (<integer uitdrukking>=1) of met 1000 baud (<integer uitdrukking>=0) geschreven worden. Wordt een file van cassette gelezen, dan schakelt de CPC464 automatisch om naar de waarde die voor die bepaalde file geldt.

Worden er cassettes van mindere kwaliteit gebruikt dan is het aan te bevelen alleen met 1000 baud te SAVEn. Verdere informatie vindt u in hoofdstuk 2.

Aanverwante instructie: **SAVE**

SQ

SQ(<kanaal>)

```
10 MODE 1
20 FOR n=20 TO 0 STEP -1
30 PRINT n;
40 SOUND 1,10+n,100,7
50 WHILE SQ(1)>127:WEND
60 NEXT
```

FUNCTIE: geeft het aantal vrije posities in de geluidsstring voor een een bepaald kanaal waarbij kanaal A=1, B=2, C=3. Tevens wordt gekeken of een kanaal actief is. Zo niet, waarom dan de eerste invoer uit de string niet wordt verwerkt. De resulterende bit is significant:

BITS 0,1,2	: aantal vrije openingen in de string.
BITS 3,4,5	: gegeven indien een RENDEVOUS status aan het begin van de geluidsstring (wachten op een ander kanaal)
BIT 6	: wordt geplaatst als de geluidsstring stilstaat.
BIT 7	: wordt geplaatst als het kanaal actief is.

Aanverwante instructies: **SOUND,ON SQ GOSUB**

SQR

SQR (<numerieke uitdrukking>)

```
PRINT SQR(9)
 3
```

FUNCTIE: geeft de vierkants-wortel uit de <numeriek uitdrukking>.

Aanverwante instructies: **PRINT**

STOP

STOP

```
300 IF n<10 THEN STOP
```

OPDRACHT: stopt het programma zodanig, dat het programma met CONT kan worden vervolgd. Het kan gebruikt worden bij het opsporen van fouten.

Aanverwante instructies: **CONT,END**

STR$

STR$ (<numerieke uitdrukking>)

```
 PRINT STR$(&766)
  1894
 PRINT STR$(&X1010100)
  84
```

FUNCTIE: zet de <numerieke uitdrukking> om in decimale <string uitdrukking>, in dezelfde vorm als gebruikt bij PRINT.

Aanverwante instructies: **VAL,PRINT,HEX$,BIN$**

STRING$

TEKST$ (<integer uitdrukking>,<teken>)

```
 PRINT STRING$(&16,"*")
  **********************
```

FUNCTIE: geeft een string, van een aantal <teken(s)> ter lengte van <integer uitdrukking>.

Aanverwante instructie: **SPACE$**

SYMBOL

SYMBOL <tekennummer>,<lijst van:<waarden>>

```
 5 MODE 2
 10 SYMBOL AFTER 90
 20 SYMBOL 93,&80,&40,&20,&10,&8,&4,&2,&1
 30 FOR n=1 TO 2000
 40 PRINT CHR$(93);
 50 NEXT
 60 GOTO 60
```

OPDRACHT: herdefinieert het <tekennummer> met nieuwe waarden. De nieuwe definitie moet van te voren met de SYMBOL AFTER instructie mogelijk gemaakt zijn. Het <tekennummer> moet tussen 0 en 255 liggen en kan een ASCII teken of een van de CPC464 tekens zijn. De <waarden> beschrijft bit-afhankelijk van boven naar beneden in de 8x8 matrix de punten van het nieuwe teken. Een 0 in het 8-bitsnummer bepaalt de kleur van de achtergrond, een 1, dat de pixel in de huidige INK wordt gezet. (zie ook aanhangsel II en III). Het voorbeeld maakt een schuine streep van links boven naar rechts onder voor de] toets.

Aanverwante instructie: **SYMBOL AFTER**

SYMBOL AFTER

SYMBOL AFTER <integer uitdrukking>

SYMBOL AFTER 90

OPDRACHT: bepaalt het aantal toegestane zelfgedefinieerde tekens. De standaard is 240, zodat 16 tekens gedefinieerd kunnen worden. Indien de <integer uitdrukking> de waarde 32 heeft, dan kunnen alle tekens met de nummers 32 tot 255 geherdefinieerd worden.
Wordt de SYMBOL AFTER instructie opnieuw gegeven dan worden de zelfgedefinieerde tekens geRESET.

Aanverwante instructie: **SYMBOL**

TAG

TAG [#<stream>]

```
10 MODE 2
11 BORDER 9
14 INK 0,12
15 INK 1,0
20 FOR n=1 TO 100
30 MOVE 200+n,320+n
40 TAG
50 IF n<70 GOTO 60 ELSE 70
60 PRINT "Tot";: GOTO 80
70 PRINT " ziens";
80 NEXT
90 GOTO 20
```

OPDRACHT: maakt het mogelijk met de PRINT instructie op de positie van de graphics-cursor te printen. Daarmee is het mogelijk tekst en symbolen naar believen met graphics te mengen. De <stream> is standaard #0, dus het beeldscherm. De linker bovenhoek van de tekens wordt op de positie van de graphics-cursor geprint. Niet afdrukbare controletekens zoals bijv. LINE FEED worden als speciale tekens geprint, indien er achter de PRINT instructie geen ; staat.

Aanverwante instructie: **TAGOFF**

TAGOFF

TAGOFF [<stream>]

TAGOFF #0

OPDRACHT: schakelt het PRINTen op de graphics-cursor positie uit. Daarna gaat de PRINT instructie terug naar de vorige tekstcursor-positie.

Aanverwante instructie: **TAG**

TAN

TAN (<numerieke uitdrukking>)

```
PRINT TAN(45)
```

FUNCTIE: geeft de TANgens van de <numerieke uitdrukking>. De <numerieke uitdrukking> mag liggen tussen de waarden -200,000 en +200,000 en wordt in RADialen gemeten, tenzij op DEG is overgeschakeld.

Aanverwante instructies: **COS,SIN,ATN,DEG,RAD**

TEST

TEST(<x-coordinaat>,<y-coordinaat>

```
PRINT TEST(300,300)
```

FUNCTIE: geeft het kleurnummer, dat op de aangegeven absolute x,y coordinaten werd gebruikt.

Aanverwante instructies: **TESTR,MOVE,MOVER,PLOT,PLOTR,DRAW,DRAWR**

TESTR

TESTR(<x-verplaatsing>,<y-verplaatsing>

```
?TEST(5,5)
```

FUNCTIE: geeft het kleurnummer, dat op de relatieve x,y coordinaat ten opzichte van de huidige graphics-cursorpositie werd gebruikt.

Aanverwante instructies: **zie ook TEST**

TIME

TIME

```
10 DATUM = INT(TIME/300)
20 TIJD = ((TIME/300)-DATUM)
30 PRINT TIJD
40 GOTO 20
```

FUNCTIE: geeft de tijd die sinds het inschakelen is verlopen, uitgezonderd de voor het lezen of het schrijven op cassette gebruikte tijd. De tijd wordt in eenheden van 1/300ste seconde gemeten.

TRON
TROFF

TRON
TROFF

OPDRACHT: TRON maakt het mogelijk de uitvoering van een programma te volgen, daar na de TRON instructie tijdens de uitvoering van het programma de regelnummers in rechte haken [] worden afgedrukt. De TROFF instructie schakelt deze functie weer uit.

Aanverwante instructie: **RUN**

UNT

UNT(<adres uitdrukking>)

```
 PRINT UNT(65535)
```

FUNCTIE: geeft de integer waarde van -32768 tot +32767 van het aangegeven 16 Bits <adres uitdrukking>.

Aanverwante instructies: **INT,FIX,CINT,ROUND**

UPPER$

UPPER$(<string uitdrukking>)

```
 PRINT UPPER$ ("schneider")
  SCHNEIDER
```

FUNCTIE: geeft de <string uitdrukking> omgezet in hoofdletters terug.

Aanverwante instructie: **LOWER$**

VAL

VAL(<string uitdrukking>)

```
 10 A$="13 is mijn ongeluks getal"
 20 PRINT VAL(A$)
```

FUNCTIE: geeft een <numerieke uitdrukking> van het begin van de <string uitdrukking>.

Aanverwante instructie: **STR$**

VPOS

VPOS(#<stream>)

PRINT VPOS(#0)

FUNCTIE: geeft de vertikale positie (=regel nummer) van de tekstcursor van de aangegeven <stream> terug. De <stream> moet in ieder geval aangegeven worden.

Aanverwante instructie: **POS**

WAIT

WAIT <poort nummer>,<masker>[,<inversie>]

WAIT &FF34,20,25

OPDRACHT: wacht met het programma tot van de I/O<poort nummer> een waarde ingelezen is die tussen 0 en 255 moet liggen. BASIC blijft tot die tijd de I/O<poort nummer> in een lus steeds weer lezen. De ingelezen waarde wordt met de <inversie> via XOR en daarna met <masker> en AND behandeld totdat het resultaat ongelijk nul is. BASIC blijft in een lus hangen met WAIT wanneer de verlangde conditie niet aanwezig is. Tikt u bovenstaand voorbeeld in, dan kunt u er alleen uitbreken via een volledige RESET van de computer.

Aanverwant instructies: **INP,OUT**

WEND

WEND

```
 10 MODE 1:REM BASIC KLOK ROUTINE
 20 INPUT "Geeft u de juiste tijd (uren, minuten, seconden)"
    ;uur,minuut,seconde
 30 CLS:datum = INT(TIME/300)
 40 WHILE uur<24
 50 WHILE minuut<60
 60 WHILE tik<60
 70 tik=(INT(TIME/300)-datum)+seconde
 80 LOCATE 70,4
 90 PRINT #0,USING "##";uur,minuut,tik
 100 WEND
 110 tik=0
 115 seconde=0
 120 minuut=minuut+1
 130 GOTO 30
 140 WEND
 150 minuut=0
 160 uur=uur+1
 170 WEND
 180 uur=1
 190 GOTO 40
```

OPDRACHT: sluit een WHILE bewerking af. De WHILE...WEND bewerking voert een deel van het programma uit zolang als aan de <logische uitdrukking> voldaan wordt. Het voorbeeld hier gebruikt de WHILE..WEND bewerking om te laten zien hoe elegant een programma opgebouwd kan worden.

Aanverwante instructie: **WHILE**

WHILE

WHILE <logische uitdrukking>

```
WHILE TAG <0
```

zie ook het WEND voorbeeld

OPDRACHT: een WHILE lus herhaalt een programmadeel zolang aan de <logische uitdrukking> voldaan wordt. WHILE staat aan het begin van de lus, WEND aan het eind.

Aanverwante instructie: **WEND**

WIDTH

WIDTH <integer uitdrukking>

```
 WIDTH 86
```

OPDRACHT: geeft het maximum aantal te printen tekens per regel op een printer aan. BASIC voert bij langere regels automatisch een carriage return uit waardoor er een nieuwe regel onstaat.

Aanverwante instructies: **PRINT,POS**

WINDOW

WINDOW [#<stream>,]<links>,<rechts>,<boven>,<onder>

```
 10 MODE 1
 20 BORDER 6
 30 WINDOW 10,30,7,18
 40 PAPER 2:PEN 3
 50 CLS
 60 PRINT CHR$(143);CHR$(242);" dit is de positie"
 70 PRINT "1,1 in het tekst venster"
 80 GOTO 80
```

OPDRACHT: bepaalt het tekstvenster (WINDOW) voor een gegeven <stream>.

Aanverwante instructie: **ORIGIN**

WINDOW SWAP

WINDOW SWAP <stream>,<stream>

```
 WINDOW SWAP 0,2
```

OPDRACHT: verwisselt de WINDOWs. Zo kunnen bijv. de normaal op <stream> #0 geprinte BASIC meldingen op een ander WINDOW gezet worden, om ze van het normale programma duidelijk te kunnen onderscheiden.

Aanverwante instructies: **WINDOW,PEN,PAPER,TAG**

WRITE

WRITE [#<stream>,][<te printen lijst>]

```
 WRITE #2,"HALLO",4,5
  "HALLO",4,5
```

OPDRACHT: voert de in de <te printen lijst> aanwezige gegevens door komma´s gescheiden uit, waarbij teksten tussen aanhalingstekens staan.

Aanverwante instructie: **PRINT**

XPOS

XPOS

```
 PRINT XPOS
```

FUNCTIE: geeft de huidige X-coordinaat van de graphics-cursor.

Aanverwante instructies: **YPOS,MOVE,MOVER,ORIGIN**

YPOS

YPOS

```
 PRINT YPOS
```

FUNCTIE: geeft de huidige Y-coordinaat van de graphics-cursor.

Aanverwante instructies: **XPOS,MOVE,MOVER,ORIGIN**

ZONE

ZONE <integer uitdrukking>

```
 10 PRINT 1,2,3
 20 ZONE 19
 30 PRINT 4,5,6
```

OPDRACHT: verandert de breedte van de printerzones die de PRINT instructie gebruikt. De standaardwaarde is 13, de nieuwe waarde kan liggen tussen 1 en 255. De zonewaarde wordt door de NEW,LOAD,CHAIN en RUN"filenaam" instructies geRESET.

Aanverwante instructies: **PRINT,WIDTH**

PRINT

PRINT [#<stream>,][<te printen lijst>][USING<formaat>][<scheider>]

<te printen lijst> is <uitdrukking>[<scheider>,<te printen gegevens>]*
<te printen gegevens> is <uitdrukking>
of
SPC(<integer uitdrukking>)
TAB(<integer uitdrukking>)

SCHRIJVEN NAAR CASSETTE

```
10 OPENOUT "DATA"
20 PRINT #9, "Hallo"
30 CLOSEOUT
```

SCHRIJVEN NAAR DE PRINTER

```
10 GETAL=23000*PI
20 PRINT #8, USING "#####.##";GETAL
30 PRINT #0, GETAL
RUN
 72256.631
Ready
 [ondertussen op de printer]
 72256.63
```

OPDRACHT: print naar de aangegeven <stream>, waarbij het aangegeven formaat gebruikt wordt (zie ook tabel Formaatbeschrijving). Schneider BASIC gebruikt de volledige industriestandaard voor de formattering met PRINT. Wordt er geen formaat aangegeven, dan print BASIC in vrij formaat. Daarbij zorgt een komma als <scheider> er voor, dat de volgende te printen <uitdrukking> in de volgende printerzone begint (standaardwaarde 13 tekens) en een ; als <scheider> dat de te printen <uitdrukking> direkt achter elkaar worden gezet.

SPC(<integer uitdrukking>) print spaties in de door <integer uitdrukking> aangegeven aantallen. Is het aantal negatief dan wordt er geen spatie geprint. Is het getal groter dan de toelaatbare breedte van de <stream>, dan wordt ze op de maximaal mogelijke waarde ingekort. SPC hoeft niet door een <scheider> afgesloten te worden, omdat altijd een ; aan het eind aangenomen wordt.

TAB(<integer uitdrukking>) geeft spaties totdat ze op de door de <integer uitdrukking> aangegeven printpositie komt. Is de <integer uitdrukking> negatief dan wordt 1 aangenomen. Is het getal groter dan de maximaal mogelijke breedte van de 'stream', dan wordt het ingekort (zie ook SPC). Is de gewenste print positie groter dan de huidige, dan wordt er tot de gewenste positie spaties geprint. Is het kleiner dan wordt een carriage return (wagen-terug,CR) uitgevoerd en worden van daaruit lege tekens geprint tot de gewenste positie bereikt is. Aangenomem wordt dat TAB altijd door ; afgesloten is.

PRINT (vervolg...)

PRINT USING "<FORMAAT>"

NUMERIEK:

Formaatteken	Beschrijving	Voorbeeld
#	Ieder # geeft een positie weer	####
.	Positie decimale punt	#.#
+	Geeft positieve getallen een + teken aan het begin of het einde van het getal	+## ##+
-	Geeft negatieve getallen het - teken aan het begin of het einde van het getal	-## ##-
**	De af te drukken getallen worden voorafgegaan door maximaal 2 asteriksen	**##.##
$$	Voor het eerste cijfer wordt een $ geprint	$$##.##
**$	Combinatie van ** en $$	**$#.##
,	Het gedeelte voor de komma wordt in groepen van drie afgedrukt, gescheiden door een komma	#,###.#
↑↑↑↑	Printen in exponentiele vorm. Daarbij wordt het getal zo gekozen dat de eerste plaats nooit nul is	#.##↑↑↑

STRING

!	Alleen het eerste teken van de tekst wordt geprint	!
<spatie>	Aantal aangegeven spaties wordt geprint plus een spatie voor en achter	< >
&	De gehele tekst wordt geprint	&

9 VERDERE PROGRAMMEER INFORMATIE

Onderwerpen in dit hoofdstuk :

* * TOELAATBARE TEKSTCURSORPOSITIES
* * CONTROLE TEKENS
* * MACHINE OPERATING SYSTEM
* * ONDERBREKINGEN

9.2 Cursorposities en controle tekens

In gebruikersprogramma's kan het voorkomen dat de tekstcursor buiten het huidige schermbereik (WINDOW) komt. Verschillende handelingen brengen de cursor weer terug naar een geldige positie. Deze zijn:

- schrijven van een teken
- printen van de cursor
- uitvoeren van de controletekens die op de volgende pagina's gekenmerkt worden met een sterretje.

Het terugbrengen van de cursor gebeurt op de volgende wijze :

a) Staat de cursor buiten de rechter rand, dan wordt het op de eerste positie van de volgende regel gezet.
b) Staat de cursor buiten de linker rand, dan wordt het op de laatste positie van de vorige regel gezet.
c) Staat de cursor boven de bovenste regel, dan wordt de gehele WINDOW een regel naar beneden gerold en de cursor in de eerste regel gezet
d) Staat de cursor onder de onderste regel, dan wordt de gehele WINDOW een regel naar boven gerold en de cursor in de laatste regel gezet.

Het testen en corrigeren vindt in de boven aangegeven volgorde plaats. De niet toegestane cursorposities kunnen de waarde 0 hebben of negatief zijn, wat betekent dat het links van of boven de WINDOW zit.

Tekenwaarden in bereik van 0...31 (zie aanhangsel III) geven geen karakter op het scherm, maar worden als controleteken geinterpreteerd (deze kunnen de computer bij verkeerd gebruik blokkeren). Een paar tekens veranderen de betekenis van de er op volgende tekens, omdat deze als parameters worden beschouwd.

Controletekens die op het grafische beeldscherm worden geprint (zie TAG instructie hfst 8), verschijnen als bijzondere tekens. Andere tekens zoals (&07 'BEL') worden bij invoer van het toetsenbord (nl **[CTRL] G**) als bijzonder teken geprint, maar behouden hun echte functie wanneer ze met PRINT CHR$ worden gebruikt (PRINT CHR$(&07)).

De met * toegekende controletekens brengen de cursor van hun bestaande- weer naar een toegestane positie, maar kunnen hem daarna in een verkeerde positie laten terugkeren. De controletekens en hun betekenis vindt u in de volgende lijst weergegeven. Eerst in HEXadecimale, daarachter in decimale vorm.

TOEGESTANE CONTROLETEKEN-OPDRACHTEN: NIET ALTIJD MET DE [CTRL]-TOETS REALISEERBAAR

Waarde			Naam	Parameter	Betekenis
&00	0		NUL		Geen invloed, wordt genegeerd.
&01	1		SOH	0..255	Drukt het teken af dat bij de parameterwaarde hoort. Zo kunnen symbolen met de waarde 0...31 afgedrukt worden.
&02	2		STX		Tekstcursor uitschakelen.
&03	3		ETX		Tekstcursor inschakelen. Opmerking: BASIC verhindert het overschrijven van de cursor; wordt alleen bij het wachten op invoer vanaf het toetsenbord opgeheven.
&04	4		EOT	0..2	Beeldschermmodus instellen. Parameter-keuze is MOD 4. Gelijk de MODE instructie.
&05	5		ENQ	0..255	Het sturen van de in de parameter aangegeven symbolen naar de graphicscursor
&06	6		ACK		Tekstscherm activeren (zie &15 NAK)
&07	7		BEL		Bel-geluid laten horen. De geluidsstring wordt leeg gemaakt.
&08	8	*	BS		Cursor 1 positie terug zetten.
&09	9	*	TAB		Cursor 1 positie vooruit zetten.
&0A	10	*	LF		Cursor 1 regel lager zetten.
&0B	11	*	VT		Cursor 1 regel hoger zetten.
&0C	12		FF		Beeldscherm wissen en de cursor in de linker bovenhoek plaatsen. Gelijk aan de CLS instructie.
&0D	13	*	CR		Cursor op de linker positie van de huidige regel zetten.
&0E	14		SO	0..15	PAPER kleur instellen. Parameter-waarde MOD 16. Gelijk aan de PAPER instructie.
&0F	15		SI	0..15	PEN kleur instellen. Parameter-waarde MOD 16. Gelijk aan de PEN instructie.
&10	16	*	DLE		Teken onder de cursor wissen. Vult teken-positie met de PAPER kleur op.

Waarde	Naam	Parameter	Betekenis
&11 17	* DC1		Regel van begin tot en met de cursor-positie wissen. De gewiste posities worden opgevuld met de PAPER kleur.
&12 18	* DC2		Regel vanaf de cursor tot aan het einde wissen. De gewiste posities worden opgevuld met de PAPER kleur.
&13 19	* DC3		Van het begin van de WINDOW tot en met de cursorpositie wissen. De gewiste posities worden opgevuld met de PAPER kleur.
&14 20	* DC4		Vanaf de cursorpositie tot het einde van de WINDOW wissen. De gewiste posities worden opgevuld met de PAPER kleur.
&15 21	NAK		Tekstscherm uitschakelen. Geen reactie totdat een ACK (&06 6) gegeven wordt.
&16 22	SYN	0..1	Transparant mode. Parameter MOD 2; met 0 uitschakelen, met 1 inschakelen.
&17 23	ETB	0..3	Graphicskleur instellen. Parameter MOD 4; (Graphics Ink Mode) 0 = Normaal 1 = ´XOR´´´ 2 = ´AND´´´ 3 = ´OR´´´
&18 24	CAN		Papier en schrijfkleur verwisselen. (PEN and PAPER Inks)
&19 25	EM	0..255 0..255 0..255 0..255 0..255 0..255 0..255 0..255 0..255	Matrix voor zelf te definieren tekens instellen. Gelijk aan SYMBOL; heeft 9 parameters nodig. De eerste geeft aan welke tekenmatrix moet worden ingesteld. De resterende acht beschrijven de matrix; de Meest Significante Bit in de eerste byte komt overeen met de bovenste linker pixel in de teken-uitvoering; de laatste bit (LSB) in de laatste byte komt overeen met de onderste rechter pixel in het teken.
&1A 26	SUB	1..80 1..80 1..25 1..25	Schermbereik (WINDOW) instellen. Gelijk aan de WINDOW instructie. Benodigd: 4 parameters. De eerste twee definieren de linker- en de rechter-randen, waarbij de kleinste de linker- en de grootste de rechter-rand bepaalt. De laatste twee parameters definieren de boven- en onder-rand. De kleinste de boven- en de grootste de onder-rand.

Waarde	Naam	Parameter	Betekenis
&1B 27	ESC		Geen invloed. Wordt genegeerd.
&1C 28	FS	0..15 0..31 0..31	Kleurenpaar voor INK instellen. Gelijk aan de INK instructie. De eerste parameter (MOD 16) definieert de PEN kleur. De beide volgende definieren de kleuren (MOD 32).
&1D 29	GS	0..31 0..31	Kleurenpaar voor BORDER instellen. Gelijk aan de BORDER instructie. De twee parameters definieren de beide kleuren (MOD 32).
&1E 30	RS		Cursor in de bovenste linkerhoek van het scherm zetten (HOME).
&1F 31	US	1..80 1..25	Cursor op de aangegeven positie zetten. Gelijk aan de LOCATE instructie. De eerste parameter definieert de kolom, de tweede de regel.

9.2 Machine Operating System
(bedrijfssysteem)

De besturing van de CPC464 wordt door een 'real time' Operating System uitgevoerd. Het Operating System bestuurt het verkeer door de computer, van invoer tot uitvoer.
Het is hoofdzakelijk de verbindingsschakel tussen de hardware en de BASIC-interpreter. Zo geeft BASIC bijv. bij wisselende kleuren de gegevens voor de kleuren, die het Operating System verwerkt. Een deel analyseert wat er moet worden gedaan en het andere controleert de timing van de handelingen.

Het Operating System van een computer wordt vaak 'Firmware' genoemd en bestaat uit Machine-Code routines, die de BASIC voor zijn instructies aanroept.

Terwijl het in BASIC bijna onmogelijk is om de CPC464 te blokkeren (anders dan de ON BREAK GOSUB instructie), is het relatief eenvoudig om met het Operating System zulke situaties te creeeren, die pas door het RESETten kunnen worden opgeheven. Daarbij moeten we wel bedenken dat dan het programma in zijn geheel verloren gaat.

Indien u wilt proberen om via de POKE instructie het interne geheugen te veranderen of met de CALL instructie aan te roepen, raden wij u aan het programma eerst te SAVEn, omdat het verloren kan gaan.

Een uitvoerige beschrijving van het Operating System van de CPC464 vindt u in de FIRMWARE specificatie (SOFT 158), dat veel verder gaat in deze materie dan deze handleiding.

9.3 Onderbrekingen

De CPC464 maakt veel gebruik van de onderbrekingsmogelijkheden van de Z80 (zgn. interrupts), waardoor een Operating System met ´multitasking´ (meerdere opdrachten worden gelijktijdig behandeld) mogelijk is. Een voorbeeld hiervan zijn de AFTER en EVERY instructies, die in hoofdstuk 8 beschreven zijn (en in hoofdstuk 10 worden). De prioriteit van de onderbrekingen zijn op de volgende manier opgebouwd :

Break [ESC][ESC]
Timer 3
Timer 2 (en de drie kanaals geluidsstring)
Timer 1
Timer 0

Onderbrekingen moeten pas ingebouwd worden als de mogelijke toestanden van de verscheidene variabelen op het tijdstip van de onderbreking goed doordacht zijn. De onderbrekings-behandelings-routines moeten ongewilde veranderingen van de variabelen in het hoofdprogramma vermijden.

De geluidsstrings hebben onafhankelijke onderbrekingen met dezelfde prioriteit. Zodra zo´n onderbreking plaats vindt, wordt ze niet door een andere nogmaals onderbroken, zodat dezelfde variabelen gebruikt kunnen worden zonder dat u zich doordacht moet opstellen tegen effecten als bij tijdsonderbrekingen bedoeld.

Zijn de geluidsstrings-onderbrekingen geactiveerd, dan treedt er direkt een onderbreking op indien de geluidsstring voor het kanaal niet vol is. Anders treedt alleen een onderbreking op, zodra de volgende toon gestart wordt en er weer plaats is in de geluidsstring. Zodra zo´n onderbreking optreedt worden de verdere toonsafhankelijke onderbrekingen geblokkeerd, zodat de behandelingsroutine de onderbrekingsmogelijkheid activeren moet, indien er nog meer onderbrekingen gewenst zijn. Wordt een toon verwerkt of de wachttoestand van de geluidsstring getest, dan worden verdere onderbrekingen geblokkeerd.

De prioriteit van de [ESC][ESC]-volgorde boven alle onderbrekingen zorgt ervoor dat een BASIC programma onderbroken kan worden, zonder dat het verloren gaat, zolang niet door een andere opdracht verhinderd wordt het programma te stoppen.

9.4 CPC464 Assembler

Voor grotere machinetaalprogramma´s heeft men een assembler nodig. De AMSOFT assembler bestaat uit een herdefinieerbare Z80 assembler met een editor, disassembler en besturingsprogramma (monitor).

10 ONDERBREKINGSMOGELIJKHEDEN

Onderwerpen in dit hoofdstuk:

* AFTER
* EVERY
* REMAIN
* De systeemklok

Mocht u het nog niet hebben gemerkt, een der belangrijkste nieuwigheden van de CPC464 is de mogelijkheid onderbrekingen (interrupts) in BASIC te realiseren. Schneider BASIC is in staat in een programma meerdere verschillende bewerkingen gelijktijdig uit te voeren. Dit wordt 'multitasking' genoemd en kan met de nieuwe instructies AFTER en EVERY worden gerealiseerd.

Deze mogelijkheid is ook duidelijk getoond in de behandeling van geluidsstrings en de RENDEVOUZ techniek. Alle tijdsafhankelijke bewerkingen worden door de systeemklok bestuurd, een quartz gestuurde timer in de rekeneenheid. Ze bewaken de tijdsafloop en stemmen alle handelingen in de rekeneenheid op elkaar af, zoals bijv. het opbouwen van het schermbeeld. Wanneer een functie in de hardware tijdsafhankelijk is wordt die door de systeemklok gestuurd.

De software kant hiervan is met de instructies AFTER en EVERY gerealiseerd. Zoals bij Schneider BASIC gewoon is, zijn deze instructies gebruikersvriendelijk. Zo wordt bijv. na de tijd (AFTER), die u in de instructieregel ingevoerd hebt, gesprongen naar een subroutine om die uit te voeren.

10.1 After

De CPC464 bezit een timer. De AFTER instructie maakt het mogelijk dat de subroutine na een geplande tijd uitgevoerd wordt. Vier onafhankelijke timers staan ter beschikking, die ieder een eigen subroutine kunnen aanspreken.

AFTER <integer>[,<integer>] GOSUB <regelnummer>

De eerste <integer> geeft aan, hoe lang het duurt voor de subroutine wordt opgeroepen. De tijd wordt in eenheden van 1/50 seconde gemeten. De tweede <integer> geeft aan, welke van de 4 timers gebruikt moet worden. De waarde moet liggen van 0...3, de standaard is 0.

Zodra de aangegeven tijd verstreken is wordt de subroutine automatisch opgeroepen, alsof een GOSUB instructie op die plaats in het programma aanwezig was. Aan het einde van de subroutine die met een normale RETURN afgesloten wordt, keert het programma naar die plaats terug, waar het onderbroken was.

De timers hebben verschillende onderbrekingsprioriteiten. Timer 3 heeft de hoogste prioriteit en Timer 0 de laagste.

AFTER instructies kunnen altijd gegeven worden, waarbij de bijbehorende subroutines en timers opnieuw worden ingesteld. De timers zijn dezelfde die ook bij de EVERY instructie worden gebruikt, zodat een AFTER instructie een vorig EVERY instructie overschrijft en ook het omgekeerde plaats vindt.

```
10 MODE 1:X=0
20 AFTER 45 GOSUB 100
30 AFTER 100,1 GOSUB 200
40 PRINT "SCHNEIDER"
50 WHILE X<100
60 LOCATE #1,30,1:PRINT #1,X:X=X+1
70 WEND
80 END
100 PRINT "randapparatuur"
110 RETURN
200 PRINT " en software"
210 RETURN
```

Bedenk wel dat bij ingebruikname van twee verschillende ´streams´ (WINDOW), het hoofdprogramma (regels 50-80) en de subroutine ieder voor zich een onafhankelijke cursor gebruiken. Nadat de tijd afgelopen is die in de instructie regel AFTER aangegeven is, springt het programma naar de overeenkomstige subroutine en voert deze dan uit.

10.2 Every

Met de EVERY instructie kan een BASICprogramma op regelmatige tijden subroutines oproepen. Vier timers staan ter beschikking, die ieder een eigen subroutine kunnen aanspreken.

```
EVERY <integer>[,<integer>] GOSUB <regelnummer>.
```

De eerste <integer> geeft aan, hoe lang moet worden gewacht voor iedere subroutine wordt aangeroepen. De tijd wordt in eenheden van 1/50 sekonden gemeten. De tweede <integer> geeft aan, welke van de 4 timers gebruikt moet worden. De waarde moet liggen van 0...3, de standaard is 0.

Zodra de aangegeven tijd verstreken is wordt de subroutine automatisch opgeroepen, alsof een GOSUB instructie op die plaats in het programma aanwezig was. Aan het einde van de subroutine die met een normale RETURN afgesloten wordt, keert het programma naar de plaats terug waar het onderbroken was.

De timers hebben verschillende onderbrekingsprioriteiten. Timer 3 heeft de hoogste prioriteit en Timer 0 de laagste.

EVERY instructies kunnen altijd gegeven worden, waarbij de bijbehorende subroutines en timers opnieuw worden ingesteld. De timers zijn dezelfde die ook bij de AFTER instructie worden gebruikt, zodat een EVERY instructie een vorige AFTER instructie overschrijft en ook het omgekeerde plaats vindt.

```
10 MODE 1:X=0
20 P100=0: EVERY 10 GOSUB 100
30 P200=0: EVERY 12,1 GOSUB 200
40 PRINT "SCHNEIDER"
50 WHILE X<200
60 LOCATE #1,30,1:PRINT #1,X:X=X+1
70 WEND
80 LOCATE 1,20:END
100 DI:PEN P100:LOCATE 1,2:PRINT "randapparatuur":EI
105 IF P100=0 THEN P100=1 ELSE P100=0
110 RETURN
200 PEN P200:LOCATE 1,3:PRINT "en software"
205 IF P200=2 THEN P200=3 ELSE P200=2
210 RETURN
```

Bemerk het gebruik van de DI en EI instructie waarmee tijds-onderbrekingen in- en uitgeschakeld kunnen worden, terwijl de tussenliggende instructies worden uitgevoerd. Hierdoor kan een routine nooit door timer 1 worden onderbroken als de tijdsvertraging van timer 1 nog loopt (regels 100 - 110) waardoor de PEN- en LOCATE instellingen voor het gebruik van print niet kunnen worden gewijzigd.

10.3 Remain

Deze functie geeft de nog overblijvende tijd van een van de 4 timers. 0 wordt gegeven indien de timer niet is ingeschakeld. Bovendien schakelt de functie de bestaande timers uit. De functie wordt als volgt gebruikt:

```
REMAIN (<integer>)
```

AANHANGSEL I

De kunst van het haalbare

Als beginneling in het programmeren wil men vaak proberen, zich aan de hand van vergelijkingen omtrend de mogelijkheden en de grenzen van een computer duidelijkheid te verschaffen. Daarom wordt in dit aanhangsel verklaard hoe en waarom er iets gebeurt.

Van start gaan

Mogelijk was voor u de tijd aangebroken om een computer te kopen om vertrouwd te raken met de computerspellen die op zo'n machine kunnen draaien. Toch zal het u misschien ook interesseren wat er onder de naam 'hardware' begrepen wordt.
Hardware is de apparatuur die u oppakt en onder uw arm kunt meenemen (computer, diskdrive, printer, plotter enz.). Hardware is alles wat niet met software wordt bedoeld: programma's, handleidingen en op cassette geladen informatie.

Bepaalde karakteristieke eigenschappen van een computer vindt men in de hardware: zoals de kleuren op de monitor of uw eigen TV-toestel. Maar dan is er software nodig om de mogelijkheden van de hardware te benutten waardoor de gewenste tekens en figuren op het beeldscherm zichtbaar worden.

De hardware zelf zorgt voor de besturing van de elektronenstraal die het beeldscherm doet oplichten. De software zorgt er voor dat alles keurig en netjes verloopt, die dus de hardware zegt wat deze moet doen en hoe. Deze zorgt voor tijdsafstemming, nauwkeurige controle en de vastlegging van de functieafloop voordat e.e.a. op het beeldscherm verschijnt.

Vraag: Wat maakt nu de ene computer beter dan de andere ?

Hard- en software zijn ieder op zich volkomen nutteloos, met de een zonder de ander kan men uberhaupt niets aanvangen. Bedenk dat uw eigen geschreven programma ook software is. Eerst de verbinding tussen hard- en software maakt het mogelijk dat de computer complexe opgaven kan verwerken. Hierna kan men aan de hand van bepaalde grondbegrippen de kwaliteit van zowel de hardware als de software overzien.

Voor Personal Computers gelden tegenwoordig de volgende criteria:

1. Het beeldscherm - het oplossend vermogen.

Het op het beeldscherm bereikbare oplossend vermogen hangt af van een combinatie van verschillende factoren, zoals het aantal kleuren die de programmeur ter beschikking staan, het aantal apart aanstuurbare beeldpunten (in de vaktaal pixel genoemd) en het aantal tekens.

Bij een vergelijk met andere computers in deze prijsklasse zult u vaststellen dat uw CPC464 op al deze punten zeer goed tevoorschijn komt.

2. De BASIC interpreter

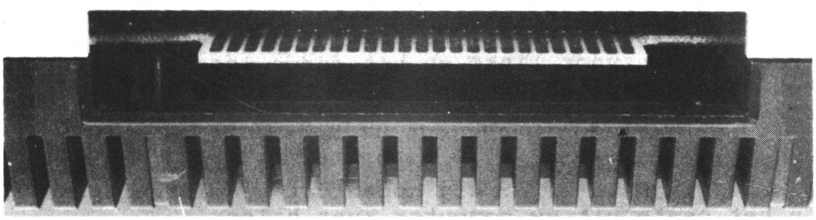
In bijna iedere computer is een BASIC interpreter ingebouwd, die het de gebruiker mogelijk maakt met relatief weinig moeite zijn eigen programma's te ontwikkelen. De programmeertaal BASIC die in het apparaat zit is zelf een geweldig gecompliceerd programma. Sinds de ontwikkeling in de Verenigde Staten werd dit met de inzet van meer dan een miljoen manjaren verder ontwikkeld. Wereldwijd is de Beginners All Purpose Symbolic Instruction Code (kortom BASIC) de meest verbreide computertaal, die net als bij spreektaal ook dialekten kent.

De in de CPC464 gebruikte versie van BASIC is een van de meest compatibele versies en BASIC programma's die voor het Disk Operating System CP/M geschreven zijn, kunnen in de meeste gevallen zonder wijzigingen draaien. Daarbij is het een snelle BASIC die in luttele sekonden een berekening uitvoert.

In de toekomst zult u vermoedelijk ook de uitdrukking machinecode tegenkomen. Dit is een vorm van opdrachten, die door de processor direkt verwerkt worden. In de machinecode gebruikt de computer zeer weinig tijd om de gestelde vraag op te lossen, meestal is dit 5 tot 15 maal sneller dan de BASIC interpreter.

Maar daar tegenover staat wel dat het 5 tot 50 maal langer kan duren om een machinecode programma te maken.

De BASIC in de Schneider computer behoort tot een van de snelste en uitgebreidste. Het heeft veel eigenschappen waarmee ervaren BASIC programmeurs de nadelen van hogere programmeertalen kunen voorkomen en mooie beeld- en geluidseffekten kan produceren.



3. Uitbreidingsmogelijkheden.

Bij de meeste computers wordt veel waarde gehecht aan de mogelijkheid om de hardware uit te breiden met joysticks, diskdrives, printer, enz. Paradoxaal genoeg hebben sommige van de meest succesvolle computers ook nog andere uitbreidingen nodig voor er normaal mee kan worden gewerkt en er op simpele wijze met eenvoudige printers of met joysticks kan worden gewerkt.

Niet altijd weet de koper wat hij in de toekomst gebruiken wil, anders is op lange termijn gezien een machine die standaard van huis uit een parallelprinter (Centronics-compatibel) en joysticks voor zijn spellen te bieden heeft goedkoper.

De CPC464 heeft ingebouwde aansluitingen voor: Centronics-printer, twee joysticks, stereo apparatuur en een uitgebreide expansiebus voor aansluiting van: diskdrives, seriele interface (RS 232), extra ROM uitbreiding enz.

Een ROM (Read Only Memory) (alleen leesbaar geheugen) is een geintegreerde schakeling die programmainformatie bezit. Ook in uw computer is een ROM ingebouwd waarin de BASIC is opgeslagen.

Voor TV-spellen gebruiken de softwarehuizen vaak insteek modulen (ROM-packs), die in plastic doosjes zijn gevat en met een aangepaste steker (natuurlijk moeten ze geschikt zijn voor de CPC464) in de machine worden geplaatst. Daarmee heeft zo´n ROM dezelfde functie als een magneetband (cassette): namelijk programmainformatie leveren.
Het grote voordeel van de ROMpack is dat de informatie die er op staat direkt uitvoerbaar is, hetgeen bij programma´s van cassette niet het geval is, daar deze enige tijd benodigen om in te laden.

Maar dat wil niet zeggen dat de gebruiker de ROMpack als opslagmedium kan gebruiken, in dat geval bent u aangewezen op cassette of diskette om de informatie te laden of te SAVEn.

Met de uitbreidingsmogelijkheden is veilig gesteld dat uw computer in de toekomst aan de meeste ontwikkelingen mee zal kunnen doen op zowel soft- als hardwaregebied. Het CPC464 systeem heeft complete en duidelijk beschreven uitbreidingsmogelijkheden.

4. Geluid en Muziek

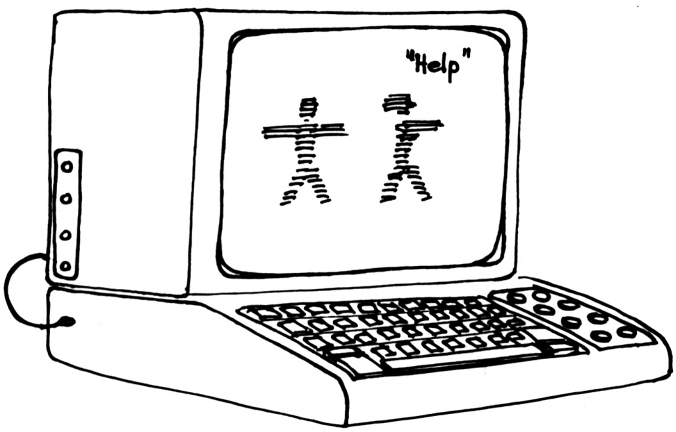
De CPC464 gebruikt een toongenerator met 3 kanalen, die een toongebied omvat van 8 octaven. Met de beinvloeding van amplitude en variatie wordt een hoge muzikale kwaliteit bereikt. Het geluid kan stereophonische effekten geven, doordat de 3 kanalen zijn verdeeld in een voor rechts, een voor links en het derde voor het acoustische midden. Met deze mogelijkheden bent u in staat om verfijnde geluidseffekten te maken.

Uiteindelijk zult u zelf moeten bepalen welke van de mogelijkheden voor u de belangrijkste zijn. Wij hopen dat u alles uitprobeert waardoor u het beste uit uw computer kunt halen.

Waarom is het niet mogelijk, dat....

Ons wordt vaak gevraagd waarom het niet mogelijk is met zo´n hoogwaardige techniek om beelden natuurgetrouw te kunnen laten zien, als te zien op de televisie.

Vraag: waarom kunnen computers niet het beeld van een wandelend persoon nabootsen, zonder dat dit altijd houterig en schokkerig moet zijn ?



Het antwoord is zowel eenvoudig als moeilijk te omschrijven. Wij moeten u zeggen, dat u niet er van uit moet gaan dat het computer-beeldscherm zo geraffineerd is als het beeldscherm van uw TV. Een TV werkt met lineare informatie, die schijnbaar met oneindig fijne verschillen tussen licht en donker werkt; waarbij ook alle kleurennuances mogelijk zijn. Dit proces zou betekenen dat omgezet in computertermen het beeldschermgeheugen van een TV 20 maal de geheugen-capaciteit nodig heeft die een huiscomputer voor zijn video-weergave gebruikt.

Dit is slechts een deel van het probleem, omdat zulke beeldbewegingen verlangen dat deze opslagcapaciteit zeer snel verwerkt moet worden (ca.50 maal per sekonden). Dit bereiken alleen machines die duizend maal duurder zijn dan uw computer. Wat dit betreft kunnen we alleen maar hopen op de toekomst.

Tot de tijd aanbreekt dat de kosten voor de snelle geheugens sterk dalen, hetgeen eens gebeurt, moeten computers met kleine geheugens werken. Doordachte hardwareplanning en goede programmaplanning kunnen er veel toe bijdragen deze situatie te verbeteren.

Wij zijn nog ver van het punt waar wij op goedkope computers realistische bewegingen en levendige beelden kunnen creeren, tenminste op dezelfde wijze als in een gemiddelde tekenfilm.

Vraag: waarom kan men niet gewoon naar de computer lopen om er eenvoudigweg een stuk tekst in te typen ?

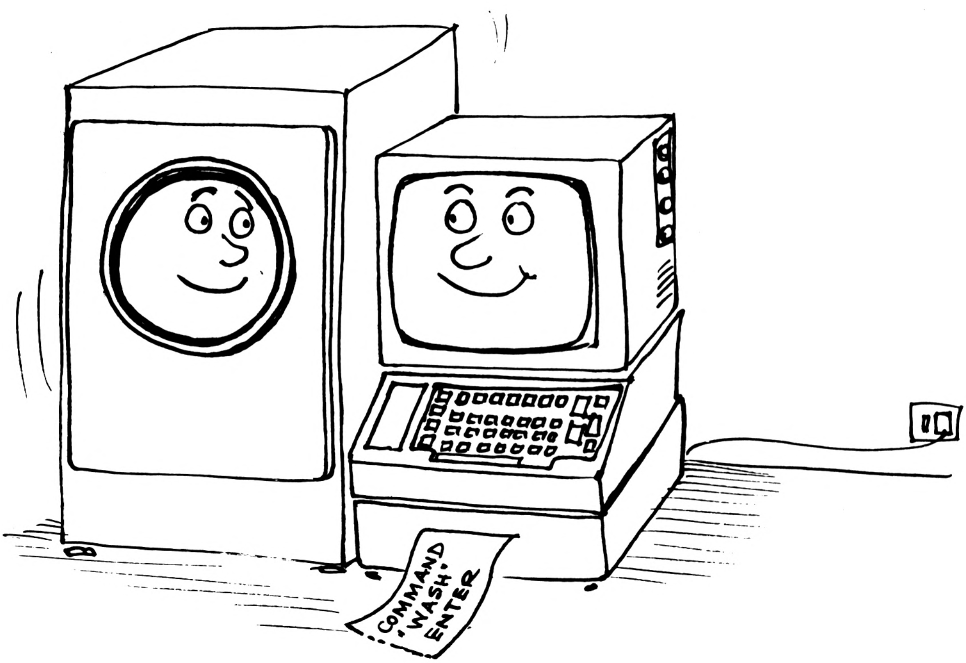
Laat u zich niet verleiden tot de gedachte dat de computer als een schrijfmachine werkt. Het beeldscherm is geen electronisch stuk papier: het geeft alleen de mogelijkheid om met de programmeertaal (en de programma's) in de computer verbinding te krijgen.

Zolang u hem niets anders zegt probeert de computer alle soorten tekens, die u via het toetsenbord intikt, als programma instructie te interpreteren. Als u op [ENTER] drukt kijkt de computer wat u hebt ingetikt om te kijken of u informatie voor de BASIC hebt ingevoerd. Is dit niet het geval, dan krijgt u de volgende melding terug:

Syntax error

Maar indien u een tekstverwerker gebruikt dan kunt u invoeren zoveel u wilt en zolang u dit maar wilt, want dan is het alsof u op electronisch papier schrijft met een electronische schrijfmachine

Het is alsof een computer al die machines verenigt, die in huis of op kantoor bekend zijn - Het televisieachtige beeldscherm, het toetsenbord, de cassetterekorder - vergeet u niet dat deze overeenkomsten volstrekt oppervlakkig zijn en dat de computer een combinatie is van vertrouwd uitziende hardware met zijn eigen persoonlijkheid.



AANHANGSEL II

Een inleiding in de achtergrond van de computertechniek

Wie heeft er angst voor de vaktaal?

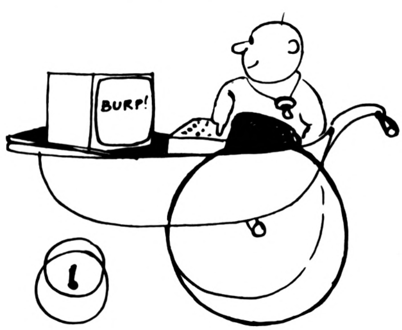
Zoals in alle gespecialiseerde industrieen is in de computertechniek een eigen vaktaal ontwikkeld, een soort kortschrift voor de overbrenging van gecompliceerde begrippen, die anders vele normale woorden nodig zouden hebben. Het zijn niet alleen de hoger technische arbeidsgebieden die zich achter de muur van termen, vaktaal en speciale begrippen verschuilen. De meeste vakgebieden waarmee we al eens mee te maken hebben gehad, hebben hun eigen vaktaal,.

Het grootste verschil is, zoals bij juristen, dat het gebruik van woorden niet zo zwaar is, maar alleen de manier waarop ze gebruikt worden. Die mensen die met computertechniek groot geworden zijn, proberen deze begrippen ongecompliceerd te gebruiken, om zo de ingewikkeldheid te verminderen.

Laat u zich niet door de eenvoudige taal in de computertechniek in verwarring brengen, het is geen literaire taal, maar een zeer precieze wetenschap. Afgezien van de SYNTAX, is de structuur zeer eenvoudig en absoluut niet verwarrend. Docenten in de computertechniek is het tot nu toe nog niet gelukt om een standaard te ontwikkelen om daarmee een programma te analyseren.

Toch kunnen programma´s geanalyseerd worden. Ook al is de betekenis van een programma niet terstond duidelijk. Tegenwoordig wordt er meer nadruk op ordelijke programmaopbouw gelegd, dan men vroeger in de microrevolutie gewoon was.

De computertechniek wordt snel door vele jongeren aangeleerd, daar het voor hen meer aankomt op de nauwkeurigheid en de eenvoudig van de ideeen die in een programma verwerkt zijn en evenzo hoe deze gegevens doorgegeven kunnen worden. U zult een tienjarige advocaat niet tegenkomen, maar een tienjarige computerfreak daarentegen wel.



Grondbeginselen van BASIC

Bijna alle huiscomputers gebruiken het meest de programmeertaal BASIC, waarmee programma's in bijna gewone taal geschreven kunnen worden. BASIC staat voor 'Beginners All-purpose Symbolic Instruction Code' (Beginners Algemene Symbolische Instructie Code), wat niet betekent dat het een primitive taal is. Vele gecompliceerde programma's worden in BASIC geschreven.

Er kan niet aan getwijfeld worden dat BASIC voor beginners is weggelegd, omdat in de wirwar van vele voorhanden zijnde programmeertalen, BASIC het meest is verbreid. Dit heeft waarschijnlijk tot verdere verbeteringen van BASIC geleid.

Van nu af worden de begrippen in de computertaal bij hun eerste optreden door vet schrift aangeduid. Hebt u geen angst voor het leren van de begrippen in dit aanhangsel.

Grondbeginselen

BASIC is een programmeertaal, die een omvang van geoorloofde instructies interpreteert en dan operaties met gegevens tijdens de programmaafloop uitvoert. Terwijl de menselijke woordenschat ca. 5 tot 8 duizend woorden omvat, komt BASIC met ca. 200 woorden uit. Programma's die in BASIC geschreven zijn, worden onderworpen aan strenge regels. De **syntax** is van zeer precieze aard, dus bij iedere poging om een normale schrijfwijze te gebruiken eindigt het programma met de koude en nuchtere melding:

Syntax error

De BASIC taal (de Syntax) is in eerste instantie bedoeld voor de verwerking van numerieke gegevens. De woorden zijn in feite rekenkundige bewerkingen zoals +/- enz. Het belangrijkste voor een beginner is te accepteren dat de computer met numerieke gegevens werkt, omdat iedere informatie als numeriek gegeven aan de **geintegreerde schakeling van de controle processor** doorgegeven wordt.

GETALLEN ALSTUBLIEFT!

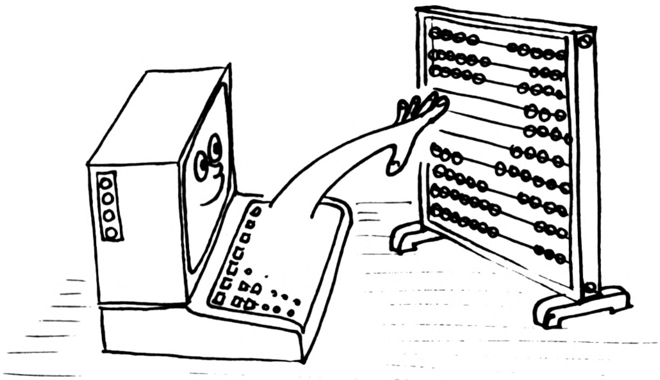
Zou een computer gebruikt worden om de complete werken van Shakespeare in het geheugen op te slaan, dan zou men bij het bekijken van het geheugen geen woord of letter terug vinden in het systeem. Ieder stuk informatie wordt omgezet in getallen, waarmee de computer wel kan werken.

BASIC interpreteert de woorden als getallen, die de computer dan met optelling en aftrekking en de eigenschappen van de **booleaanse logica** behandelen kan, waarbij het de gegevens vergelijkt en enige delen daarvan uitzoekt, om bijv. te controleren of een getal groter is dan een ander of om een bepaalde afloop in te leiden, indien het ene of het andere getal dit verlangt.

De computer werkt met de eenvoudige opdrachten: "aan" en "uit".

Het vermenigvuldigen wordt in de vorm van meerdere optellingen uitgevoerd. De BASIC opdracht: vermenigvuldig 35 met 10 (35*10), wordt omgezet in 10 keer 35 op te tellen.

In een gebied van de **Centrale Processor Unit (CPU)** wordt het getal 10 opgeslagen, in een ander het getal 35. Als 35 bij zich zelf opgeteld wordt, wordt het gebied met de inhoud 10 met 1 verminderd totdat het 0 wordt. De vermenigvuldiging eindigt nu met het resultaat 350 in een ander gebied van de CPU, vanwaar het gegeven antwoord als **output** wordt gegeven.



Als dit proces u te omslachtig voorkomt, dan heeft u gelijk, maar u hebt het eerste en belangrijkste deel van de computertechniek ondekt. Dus BASIC interpreteert opdrachten in programma's en vormt deze om in een taal die de CPU behandelen kan. De computer verstaat alleen twee begrippen: JA of NEE, in binaire schrijfwijze: 1 of 0. En het geeft dit in booleaanse logica weer als 'waar' of 'niet waar'.

Het omschakelen tussen deze beide toestanden is de wezenlijke betekenis van de digitale techniek en staat bekend als **toggling**.
In de natuur bewegen processen geleidelijk van de ene stabiele toestand naar de andere. Met andere woorden, de overgang doorloopt een lange weg tussen beide toestanden. In een ideale digitale omgeving kost zo'n omschakeling absoluut geen tijd, maar omdat halfgeleidertechnieken aangeven dat er enige vertraging plaatsvindt, genoemd als verspreidingsvertraging (propagation delays). De som van deze vertragingen is de reden waarom het wat tijd kost, voordat de informatie bewerkt is en er een antwoord gegeven kan worden.

In ieder geval moet een computer een bepaalde tijd rekenen voor een opgave klaar is en de gegevens daarvan voor verdere bewerking doorgegeven kunnen worden. Een digitaal proces is of zwart of wit, waarbij de overgang via grijstonen geen betekenis heeft. Een vloeiende overgang voert door alle grijstonen.

Wanneer het uiteindelijke antwoord of 0 of 1 kan zijn, dan is er geen mogelijkheid om bijna correct te zijn. Dat de computer soms fouten maakt bij behandeling van numerieke gegevens, is een gevolg van het feit dat de computer in de grootte van een getal begrensd is. Numerieke gegevens die deze grens overschrijden worden afgesneden, waarbij het gevolg kan zijn dat er afrondingsfouten optreden. Daardoor kan 999.999.999 het getal 1.000.000.000 worden.

Bits en Bytes

We zijn gewend om met getallen in decimale vorm om te gaan, waarbij de basis het getal 10 is: de waarden worden met de cijfers van 0 tot 9 uitgebeeld. Het getallenstelsel met de cijfers 0 en 1 is het binaire stelsel, en de eenheden waarmee het stelsel werkt heten **bits** **(BInary digiT)**. Het kan minder verwarrend zijn als u 0 en 1 in samenhang met het decimale systeem vergeet en daarvoor in de plaats symbolen voor de functies neemt, bijv.: fl en *, zolang er maar twee situaties mee worden aangeduid.

De samenhang tussen bits en de decimale schrijfwijze is eenvoudig te begrijpen.

Het is gebruikelijk de lengte van binaire cijfers aan te geven en de leidende nullen te schrijven, zodat altijd de complete aantallen bits gebruikt worden.
Decimaal 7 wordt 00111 binair in de 5 bits schrijfwijze.

In het binaire systeem kunnen de cijfers als posities in kolommen voorgesteld worden, die bepalen of er machten van 2 voorhanden zijn (1) of niet (0).

$2^0 = 1$
$2^1 = 2 = 2 = 2(2^0)$
$2^2 = 4 = 2*2 = 2(2^1)$
$2^3 = 8 = 2*2*2 = 2(2^2)$
$2^4 = 16 = 2*2*2*2 = 2(2^3)$

Zodat de posities er als volgt uit zien.

2^4	2^3	2^2	2^1	2^0
1	0	0	1	1
(16	0	0	2	1) = 19

Een korte schrijfwijze om binaire cijfers te maken is het begrip byte (1 byte is 8 bits). Het grootste getal dat in een byte kan worden opgeslagen is binair 11111111 of decimaal 255. Dat geeft 256 echte variaties 00000000 inbegrepen (0 gebruikt de computer ook).

De computer behandelt de gegevens in hoeveelheden van 8 bits. 256 is geen groot getal, zodat bij adressering van het geheugen 2 bytes gebruikt worden. Het geheugen kan in een rij- en kolomsysteem (array) opgedeeld worden, waarbij ieder element op de volgende wijze kan worden gelocaliseerd:

0	1	2	3	4	5	6	7	8	9
1									
2									
3									
4									
5			1		1				
6									
7									
8									
9									

Deze matrix kan tot 10 * 10 informatieeenheden opnemen, waarvan de adressen ieder in het bereik van 0 tot 9 liggen. Op positie 4,5 is een 1 opgeslagen, evenals op 6,5.

Een gebied met 256 * 256 eenheden omvat 65536 enkele locaties, die met 8-bits adressen voor de horizontale en vertikale assen aangesproken kunnen worden. Met onze 0 en 1 kunnen we nu 65536 elementen adresseren.

Het volgende punt is de korte schrijfwijze van **Kilo**byte (kByte of eenvoudig K), wat 1024 bytes betekent. 1024 is het decimale begrip voor kilo, het dichtsbijzijnde binaire veelvoud van 10 en verklaart, waarom een computer met een 64K geheugen in werkelijkheid met 65536 bytes (64 * 1024) uitgerust is.

Gelukkig voert de BASICinterpreter alle noodzakelijke conversies voor u uit, zodat u op eenvoudige wijze programmeren kunt, zonder het gehele binaire stelsel te kennen. Mocht het zijn dat u het binaire stelsel begrijpt, dan zal het u helpen sneller mysteries van significante getallen te begrijpen, die onvermijdelijk zullen opduiken, wanneer u zich in de wetenschap van de computertechniek inwerkt.

Het loont de moeite het binaire stelsel te begrijpen en de verschillende getallen zoals 255 , 1024 , enz. te herkennen, omdat het zeer onwaarschijnlijk is dat deze grondbeginselen in de computerwereld binnen afzienbare tijd veranderen

Maar toch...

De binaire schrijfwijze is eenvoudig en elegant, maar ze is lastig te schrijven en leidt tot fouten bij het lezen. Het binaire stelsel is met andere getallenstelsels verwant die dienen als korte schrijfwijze voor het programmeren. Zo´n getallenstelsel, dat bij computersystemen gebruikt wordt, heet Hex (van **hexadecimaal).**

Daarbij zijn de getallen gebaseerd op het getal 16 en worden met een enkel teken uitgebeeld.

Decimaal

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Hex

0 1 2 3 4 5 6 7 8 9 A B C D E F

Het hexadecimale systeem kan de 8 bits in een byte in twee blokken van ieder vier bits verdelen, daar 15 een vierbits getal is: 1111 binair. Het eerste blok geeft het aantal complete eenheden van 15 aan, het tweede blok het deel van het laatste blok. Hier blijkt het elegante van de binaire schrijfwijze.

Hier nogmaals de tabel voor de invoering van de binaire schrijfwijze:

Dezimal	Binär	CPC464isch	Hexadezimal
0	0	fl	0
1	1	*	1
2	10	*fl	2
3	11	**	3
4	100	*flfl	4
5	101	*fl*	5
6	110	**fl	6
7	111	***	7
8	1000	*flflfl	8
9	1001	*flfl*	9
10	1010	*fl*fl	A
11	1011	*fl**	B
12	1100	**flfl	C
13	1101	**fl*	D
14	1110	***fl	E
15	1111	****	F
16	10000	*flflflfl	10

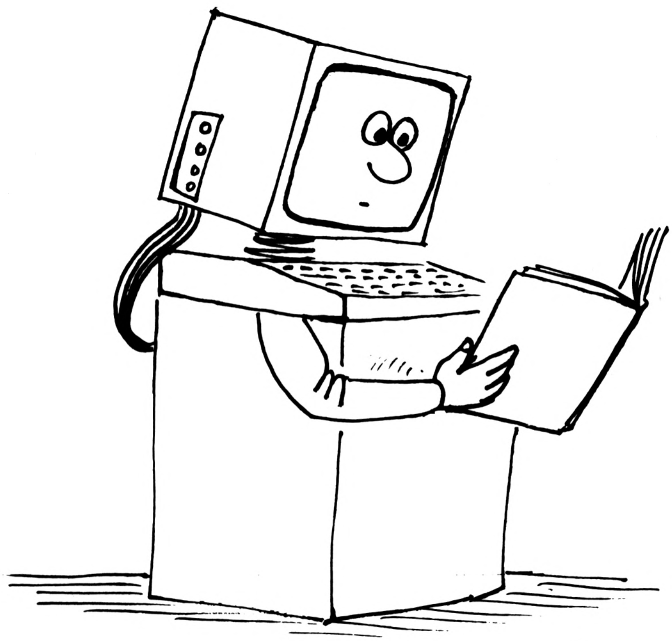
Een 8 bits getal 11010110 kan in twee 4 bits getallen (half-bytes of ´nibbles´) onderverdeeld worden:HEX D6. In dit handboek worden Hex getallen altijd met het symbool & aangegeven en dit getallen-stelsel wordt ook door de meeste programmeurs in assembler gebruikt. Assembler biedt het nauwste contact met machinecode, omdat het in eenvoudige alfabetische tekens (mnemonics) de echte machinecode getallen kan weergeven.

Wanneer u met Hex werkt, moet u eerst het aantal blokken van 16 bepalen, dan het tweede cijfer daarbij optellen, om dan definitief de decimale waarde te berekenen. Men is geneigd om het getal &D6 als 13+6 of 136 te beschouwen, maar goed is: (13*16)+(16)=214.

Het is hetzelfde bij een decimaal stelsel, waarin men bijv. voor 89 leest (8 * 10)+(9). De vermenigvuldiging met 10 komt u eenvoudig voor, omdat u hiermee vaker te doen heeft.

Als u zover bent gekomen zonder door de bomen het bos niet meer te zien, dan heeft u een goede start gemaakt in de grondbeginselen van een computer. En als u zich afvraagt waarom dit alles, dan heeft u gelijk. Een computer is een machine die met zeer eenvoudige ideeen en concepten werkt, maar dan wel zeer snel (miljoenste van een seconde) en met een enorme capaciteit, om uit te maken, welke gegevens naar het gevraagde antwoord leiden.

Om de theorie van uw computer verder te begrijpen staan er verschillende boeken ter beschikking. Sommige zullen u alleen maar verwarren, andere daarentegen zullen u veel verder brengen en u de eenvoud onthullen alsmede de fundamentele relatie die er bestaat tussen de getalstelsels en de behandelingswijze van uw computer.



AANHANGSEL III

DE ASCII TEKENS EN DE GRAFISCHE TEKENS VAN DE CPC464

III.1 ASCII

Hieronder is de volledige standaard ASCII-tabel afgedrukt, waardoor het nazoeken van gegevens makkelijker wordt gemaakt. Gegeven zijn de decimale, octale en hexadecimale notatie van de diverse tekens tekens en waar mogelijk het ASCII teken. Ook alle CPC464 tekencellen zijn in detail gegeven.

DEC	OCTAL	HEX	ASCII characters
0	000	00	NUL ([CTRL]@)
1	001	01	SOH ([CTRL]A)
2	002	02	STX ([CTRL]B)
3	003	03	ETX ([CTRL]C)
4	004	04	EOT ([CTRL]D)
5	005	05	ENQ ([CTRL]E)
6	006	06	ACK ([CTRL]F)
7	007	07	BEL ([CTRL]G)
8	010	08	BS ([CTRL]H)
9	011	09	HT ([CTRL]I)
10	012	OA	LF ([CTRL]J)
11	013	OB	VT ([CTRL]K)
12	014	OC	FF ([CTRL]L)
13	015	OD	CR ([CTRL]M)
14	016	OE	SO ([CTRL]N)
15	017	OF	SI ([CTRL]O)
16	020	10	DLE ([CTRL]P)
17	021	11	DC1 ([CTRL]Q)
18	022	12	DC2 ([CTRL]R)
19	023	13	DC3 ([CTRL]S)
20	024	14	DC4 ([CTRL]T)
21	025	15	NAK ([CTRL]U)
22	026	16	SYN ([CTRL]V)
23	027	17	ETB ([CTRL]W)
24	030	18	CAN ([CTRL]X)
25	031	19	EM ([CTRL]Y)
26	032	1A	SUB ([CTRL]Z)
27	033	1B	ESC
28	034	1C	FS
29	035	1D	GS
30	036	1E	RS
31	037	1F	US
32	040	20	SP
33	041	21	!
34	042	22	"
35	043	23	#
36	044	24	$
37	045	25	%
38	046	26	&
39	047	27	'
40	050	28	(
41	051	29	)
42	052	2A	*
43	053	2B	+
44	054	2C	,
45	055	2D	-
46	056	2E	.
47	057	2F	/
48	060	30	Ø
49	061	31	1

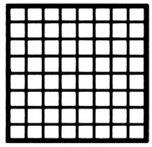
DEC	OCTAL	HEX	ASCII
50	062	32	2
51	063	33	3
52	064	34	4
53	065	35	5
54	066	36	6
55	067	37	7
56	070	38	8
57	071	39	9
58	072	3A	:
59	073	3B	;
60	074	3C	<
61	075	3D	=
62	076	3E	>
63	077	3F	?
64	100	40	@
65	101	41	A
66	102	42	B
67	103	43	C
68	104	44	D
69	105	45	E
70	106	46	F
71	107	47	G
72	110	48	H
73	111	49	I
74	112	4A	J
75	113	4B	K
76	114	4C	L
77	115	4D	M
78	116	4E	N
79	117	4F	O
80	120	50	P
81	121	51	Q
82	122	52	R
83	123	53	S
84	124	54	T
85	125	55	U
86	126	56	V
87	127	57	W
88	130	58	X
89	131	59	Y
90	132	5A	Z
91	133	5B	[
92	134	5C	\
93	135	5D	]
94	136	5E	↑
95	137	5F	_
96	140	60	`
97	141	61	a
98	142	62	b
99	143	63	c

DEC	OCTAL	HEX	ASCII
100	144	64	d
101	145	65	e
102	146	66	f
103	147	67	g
104	150	68	h
105	151	69	i
106	152	6A	j
107	153	6B	k
108	154	6C	l
109	155	6D	m
110	156	6E	n
111	157	6F	o
112	160	70	p
113	161	71	q
114	162	72	r
115	163	73	s
116	164	74	t
117	165	75	u
118	166	76	v
119	167	77	w
120	170	78	x
121	171	79	y
122	172	7A	z
123	173	7B	{
124	174	7C	¦
125	175	7D	}
126	176	7E	~

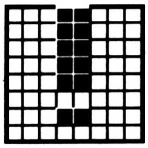
ASCII codes and CPC464 characters

III.2 CPC464 MACHINE-SPECIFIEKE FIRMWARE TEKENSET

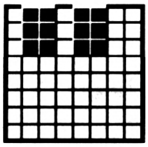
De hier afgebeelde tekens zijn in een standaard 8 * 8 matrix uitgebeeld zoals ze op het beeldscherm verschijnen. Zelf te definieren tekens kunnen voor speciale effekten samengesteld worden en tegen elkaar aansluiten. Zie ook de SYMBOL instructiebeschrijving in hoofdstuk 8.



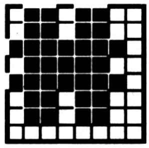
32 &H20
&X00100000



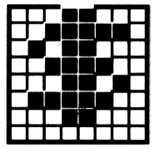
33
&H21
&X00100001



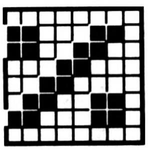
34
&H22
&X00100010



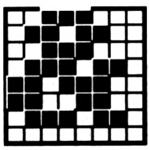
35
&H23
&X00100011



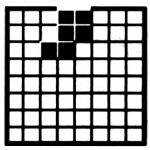
36
&H24
&X00100100



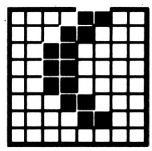
37
&H25
&X00100101



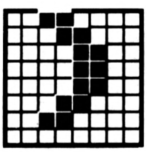
38
&H26
&X00100110



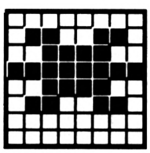
39
&H27
&X00100111



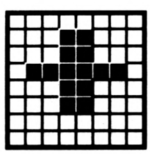
40
&H28
&X00101000



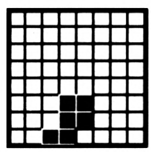
41
&H29
&X00101001



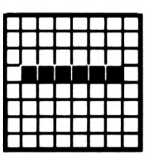
42
&H2A
&X00101010



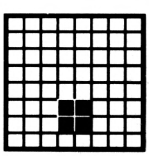
43
&H2B
&X00101011



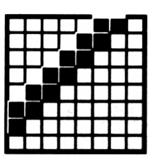
44
&H2C
&X00101100



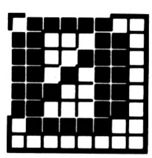
45
&H2D
&X00101101



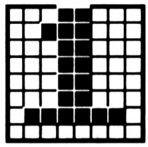
46
&H2E
&X00101110



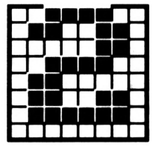
47
&H2F
&X00101111



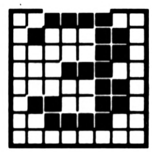
48
&H30
&X00110000



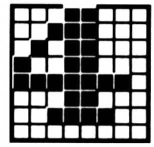
49
&H31
&X00110001



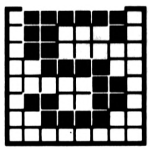
50
&H32
&X00110010



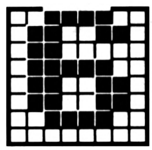
51
&H33
&X00110011



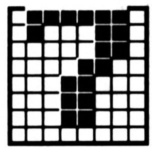
52
&H34
&X00110100



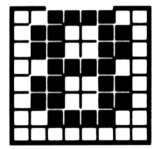
53
&H35
&X00110101



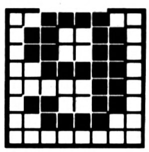
54
&H36
&X00110110



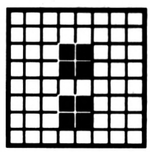
55
&H37
&X00110111



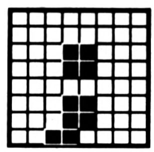
56
&H38
&X00111000



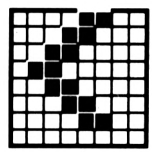
57
&H39
&X00111001



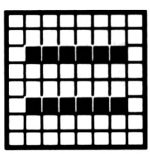
58
&H3A
&X00111010



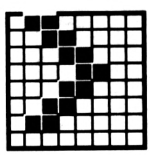
59
&H3B
&X00111011



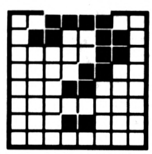
60
&H3C
&X00111100



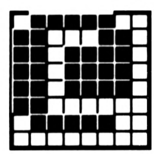
61
&H3D
&X00111101



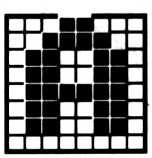
62
&H3E
&X00111110



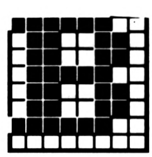
63
&H3F
&X00111111



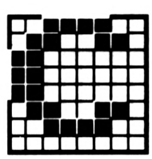
64
&H40
&X01000000



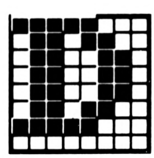
65
&H41
&X01000001



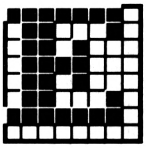
66
&H42
&X01000010

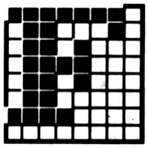


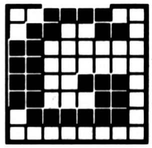
67
&H43
&X01000011

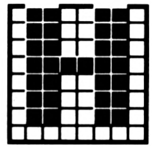


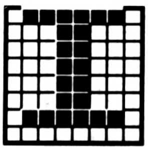
68
&H44
&X01000100

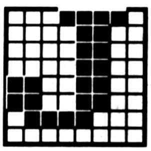

69
&H45
&X01000101

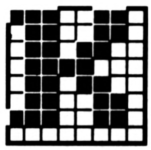

70
&H46
&X01000110

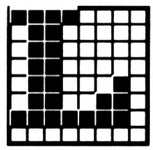

71
&H47
&X01000111

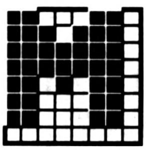

72
&H48
&X01001000

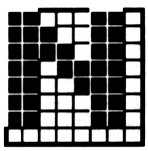

73
&H49
&X01001001

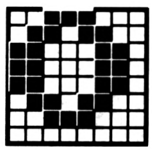

74
&H4A
&X01001010

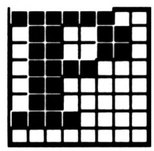

75
&H4B
&X01001011

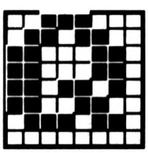

76
&H4C
&X01001100

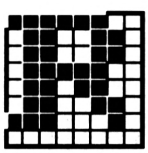

77
&H4D
&X01001101

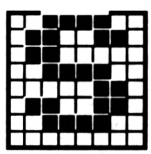

78
&H4E
&X01001110

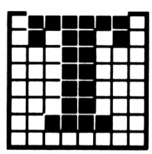

79
&H4F
&X01001111

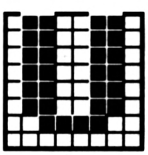

80
&H50
&X01010000

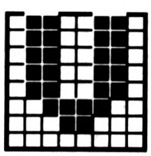

81
&H51
&X01010001

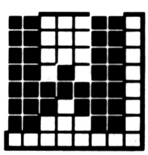

82
&H52
&X01010010

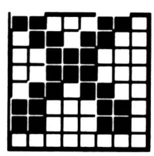

83
&H53
&X01010011

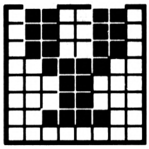

84
&H54
&X01010100

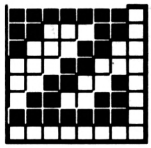

85
&H55
&X01010101

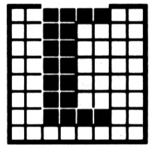

86
&H56
&X01010110

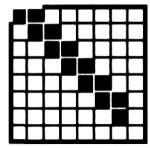

87
&H57
&X01010111

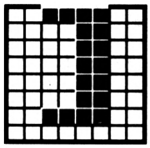

88
&H58
&X01011000

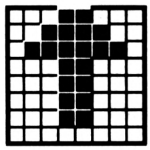

89
&H59
&X01011001

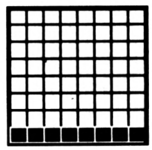

90
&H5A
&X01011010

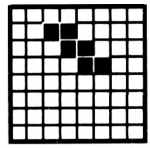

91
&H5B
&X01011011

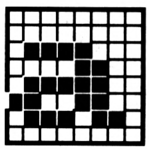

92
&H5C
&X01011100

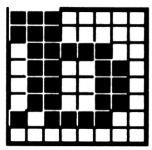

93
&H5D
&X01011101

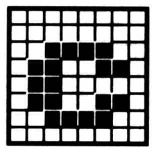

94
&H5E
&X01011110

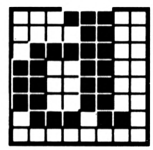

95
&H5F
&X01011111

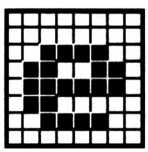

96
&H60
&X01100000

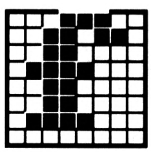

97
&H61
&X01100001

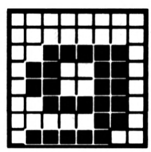

98
&H62
&X01100010

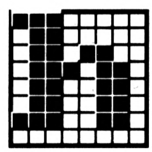

99
&H63
&X01100011

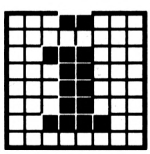

100
&H64
&X01100100

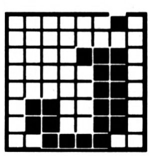

101
&H65
&X01100101

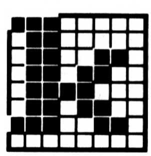

102
&H66
&X01100110

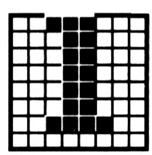

103
&H67
&X01100111

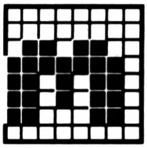

104
&H68
&X01101000


105
&H69
&X01101001

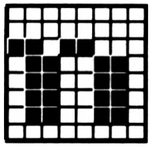

106
&H6A
&X01101010


107
&H6B
&X01101011

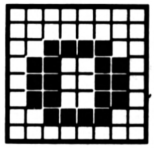

108
&H6C
&X01101100



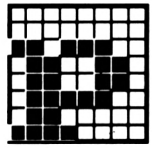
109
&H6D
&X01101101



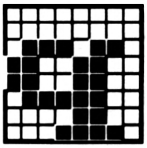
110
&H6E
&X01101110



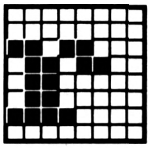
111
&H6F
&X01101111



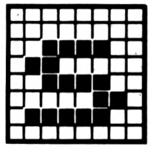
112
&H70
&X01110000



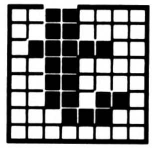
113
&H71
&X01110001



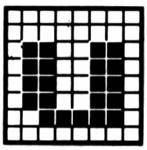
114
&H72
&X01110010



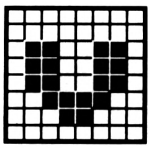
115
&H73
&X01110011



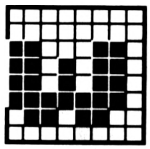
116
&H74
&X01110100



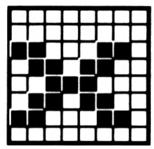
117
&H75
&X01110101



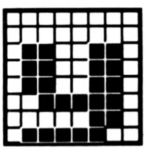
118
&H76
&X01110110



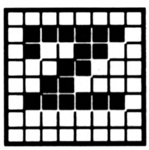
119
&H77
&X01110111



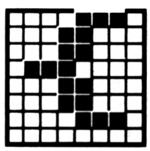
120
&H78
&X01111000



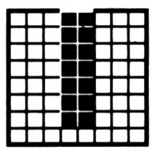
121
&H79
&X01111001



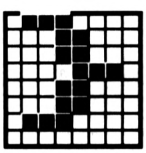
122
&H7A
&X01111010



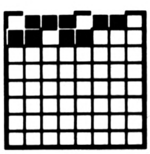
123
&H7B
&X01111011



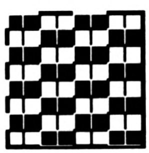
124
&H7C
&X01111100



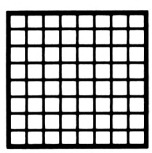
125
&H7D
&X01111101



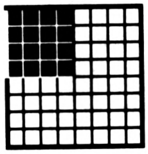
126
&H7E
&X01111110



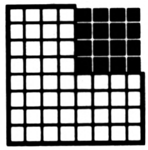
127
&H7F
&X01111111



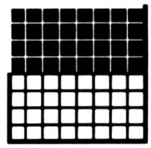
128
&H80
&X10000000



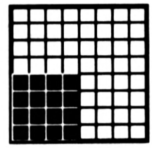
129
&H81
&X10000001



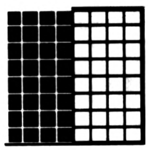
130
&H82
&X10000010



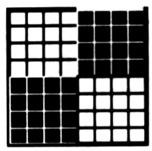
131
&H83
&X10000011



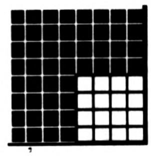
132
&H84
&X10000100



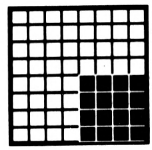
133
&H85
&X10000101



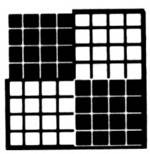
134
&H86
&X10000110



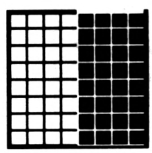
135
&H87
&X10000111



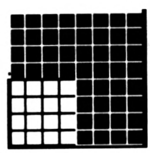
136
&H88
&X10001000



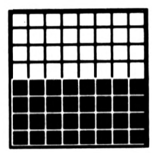
137
&H89
&X10001001



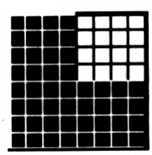
138
&H8A
&X10001010



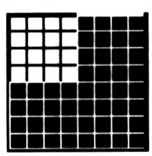
139
&H8B
&X10001011



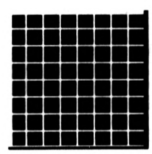
140
&H8C
&X10001100



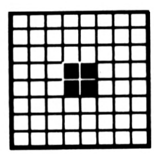
141
&H8D
&X10001101



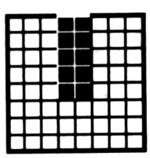
142
&H8E
&X10001110



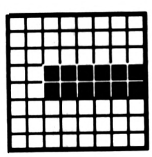
143
&H8F
&X10001111



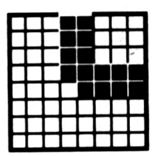
144
&H90
&X10010000



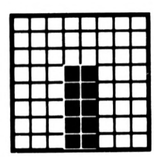
145
&H91
&X10010001



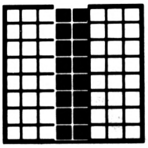
146
&H92
&X10010010

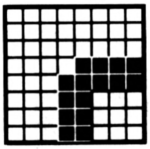


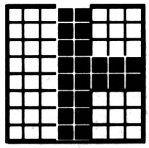
147
&H93
&X10010011

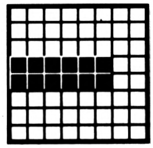


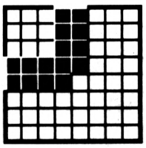
148
&H94
&X10010100

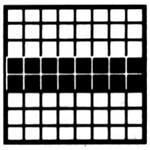

149
&H95
&X10010101

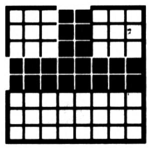

150
&H96
&X10010110

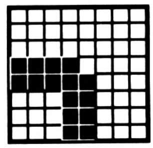

151
&H97
&X10010111

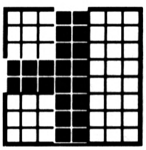

152
&H98
&X10011000

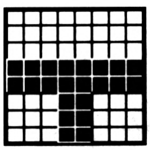

153
&H99
&X10011001

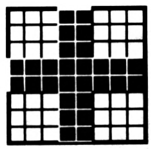

154
&H9A
&X10011010

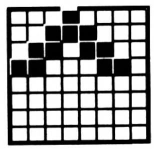

155
&H9B
&X10011011

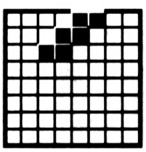

156
&H9C
&X10011100

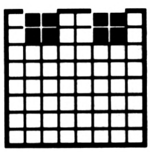

157
&H9D
&X10011101

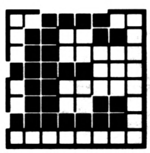

158
&H9E
&X10011110

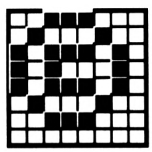

159
&H9F
&X10011111

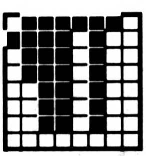

160
&HA0
&X10100000

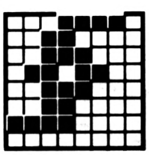

161
&HA1
&X10100001

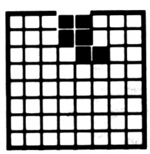

162
&HA2
&X10100010

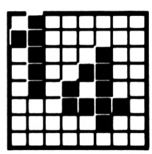

163
&HA3
&X10100011

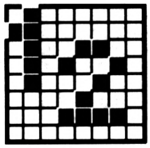

164
&HA4
&X10100100

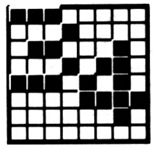

165
&HA5
&X10100101

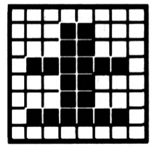

166
&HA6
&X10100110

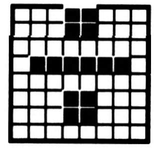

167
&HA7
&X10100111

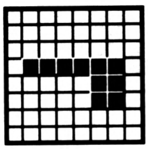

168
&HA8
&X10101000

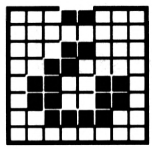

169
&HA9
&X10101001

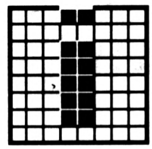

170
&HAA
&X10101010

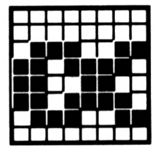

171
&HAB
&X10101011

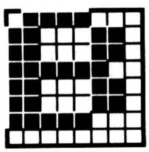

172
&HAC
&X10101100

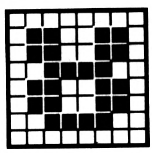

173
&HAD
&X10101101

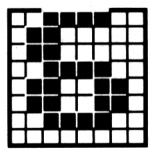

174
&HAE
&X10101110

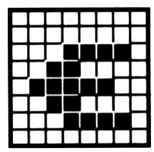

175
&HAF
&X10101111

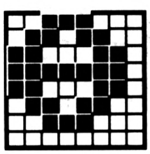

176
&HB0
&X10110000

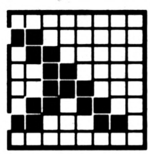

177
&HB1
&X10110001

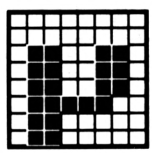

178
&HB2
&X10110010

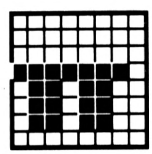

179
&HB3
&X10110011

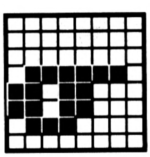

180
&HB4
&X10110100

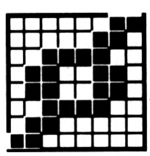

181
&HB5
&X10110101

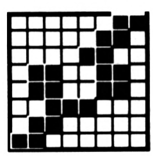

182
&HB6
&X10110110

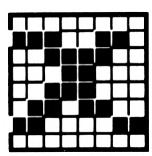

183
&HB7
&X10110111

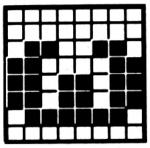

184
&HB8
&X10111000

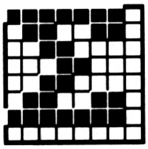

185
&HB9
&X10111001

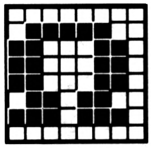

186
&HBA
&X10111010

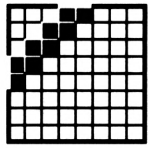

187
&HBB
&X10111011

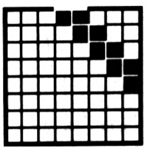

188
&HBC
&X10111100

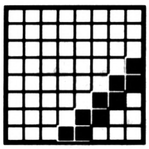

189
&HBD
&X10111101

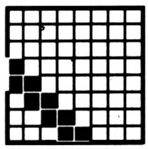

190
&HBE
&X10111110

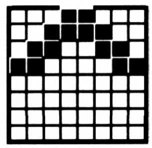

191
&HBF
&X10111111

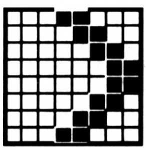

192
&HC0
&X11000000

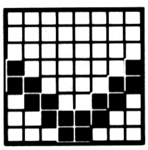

193
&HC1
&X11000001

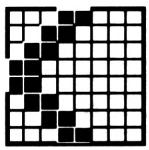

194
&HC2
&X11000010

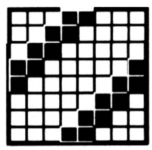

195
&HC3
&X11000011

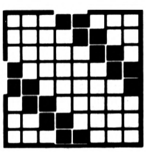

196
&HC4
&X11000100

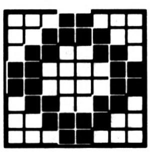

197
&HC5
&X11000101

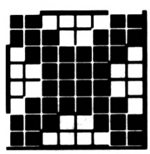

198
&HC6
&X11000110

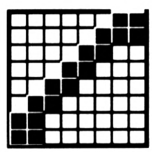

199
&HC7
&X11000111

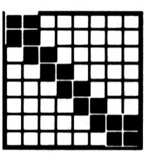

200
&HC8
&X11001000

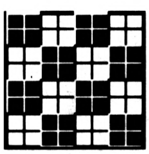

201
&HC9
&X11001001

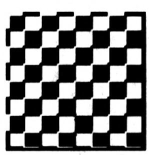

202
&HCA
&X11001010

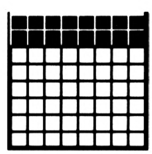

203
&HCB
&X11001011

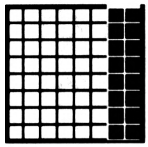

204
&HCC
&X11001100

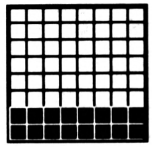

205
&HCD
&X11001101

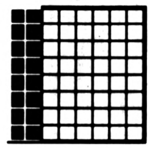

206
&HCE
&X11001110

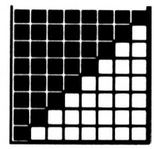

207
&HCF
&X11001111

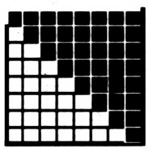

208
&HD0
&X11010000

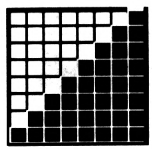

209
&HD1
&X11010001

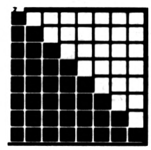

210
&HD2
&X11010010

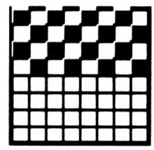

211
&HD3
&X11010011

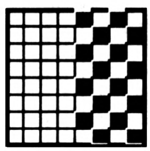

212
&HD4
&X11010100

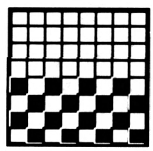

213
&HD5
&X11010101

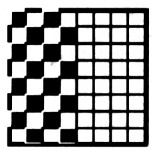

214
&HD6
&X11010110

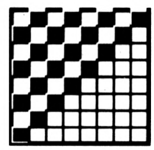

215
&HD7
&X11010111

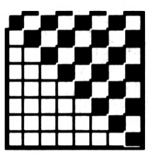

216
&HD8
&X11011000

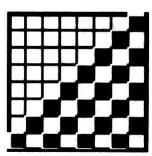

217
&HD9
&X11011001

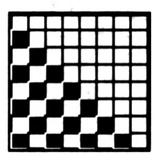

218
&HDA
&X11011010

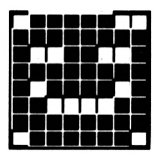

219
&HDB
&X11011011

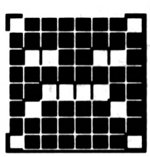

220
&HDC
&X11011100

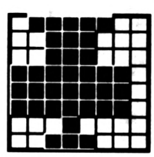

221
&HDD
&X11011101

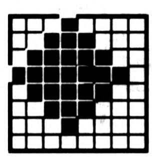

222
&HDE
&X11011110

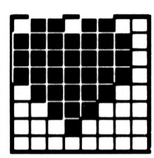

223
&HDF
&X11011111

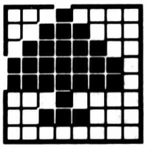

224
&HE0
&X11100000

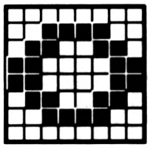

225
&HE1
&X11100001

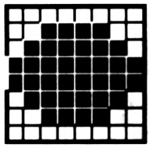

226
&HE2
&X11100010

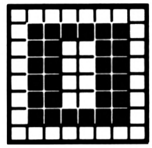

227
&HE3
&X11100011

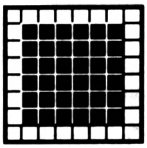

228
&HE4
&X11100100

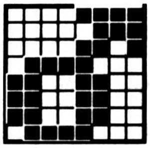

229
&HE5
&X11100101

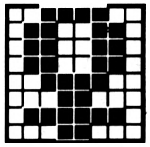

230
&HE6
&X11100110

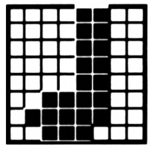

231
&HE7
&X11100111

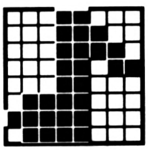

232
&HE8
&X11101000

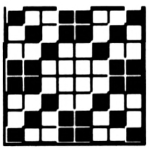

233
&HE9
&X11101001

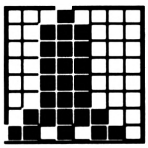

234
&HEA
&X11101010

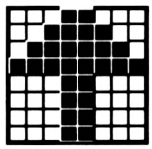

235
&HEB
&X11101011

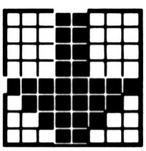

236
&HEC
&X11101100

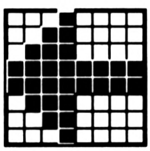

237
&HED
&X11101101

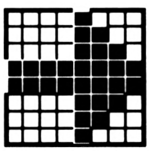

238
&HEE
&X11101110

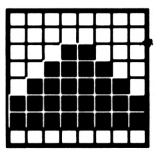

239
&HEF
&X11101111

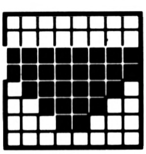

240
&HF0
&X11110000

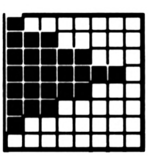

241
&HF1
&X11110001

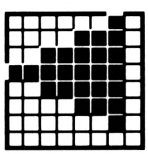

242
&HF2
&X11110010

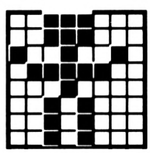

243
&HF3
&X11110011

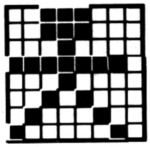

244
&HF4
&X11110100


245
&HF5
&X11110101

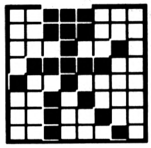

246
&HF6
&X11110110


247
&HF7
&X11110111

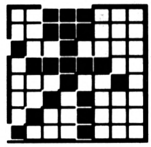

248
&HF8
&X11111000



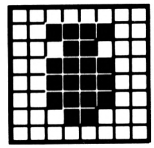
249
&HF9
&X11111001



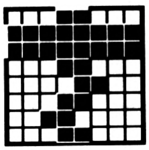
250
&HFA
&X11111010



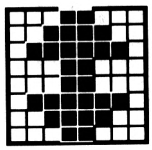
251
&HFB
&X11111011



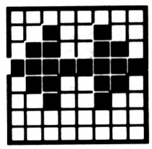
252
&HFC
&X11111100



253
&HFD
&X11111101

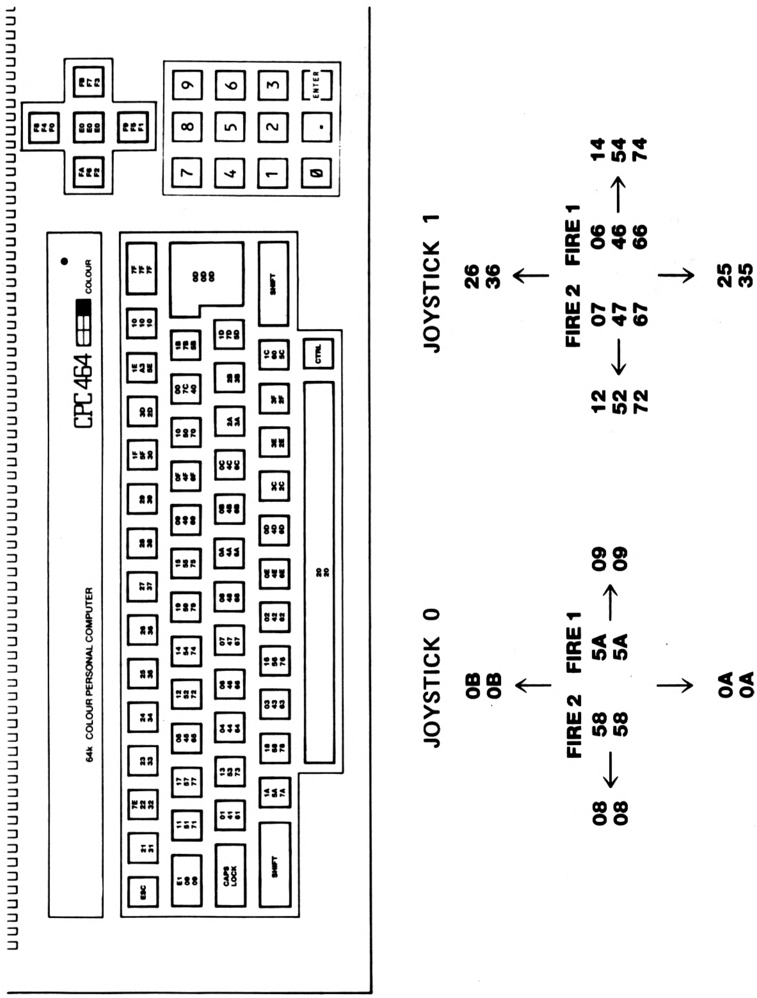


254
&HFE
&X11111110

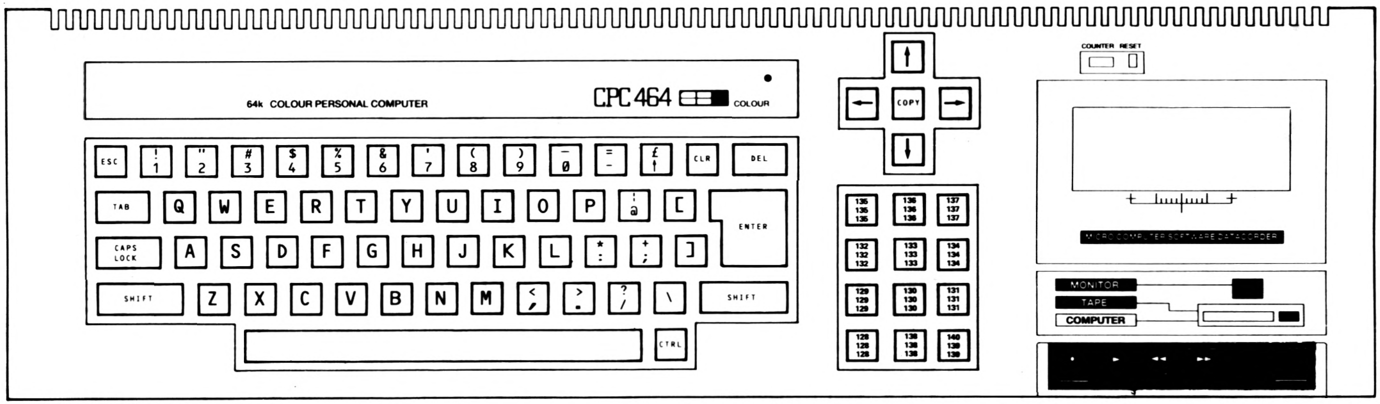


255
&HFF
&X11111111

STANDAARD ASCII WAARDEN



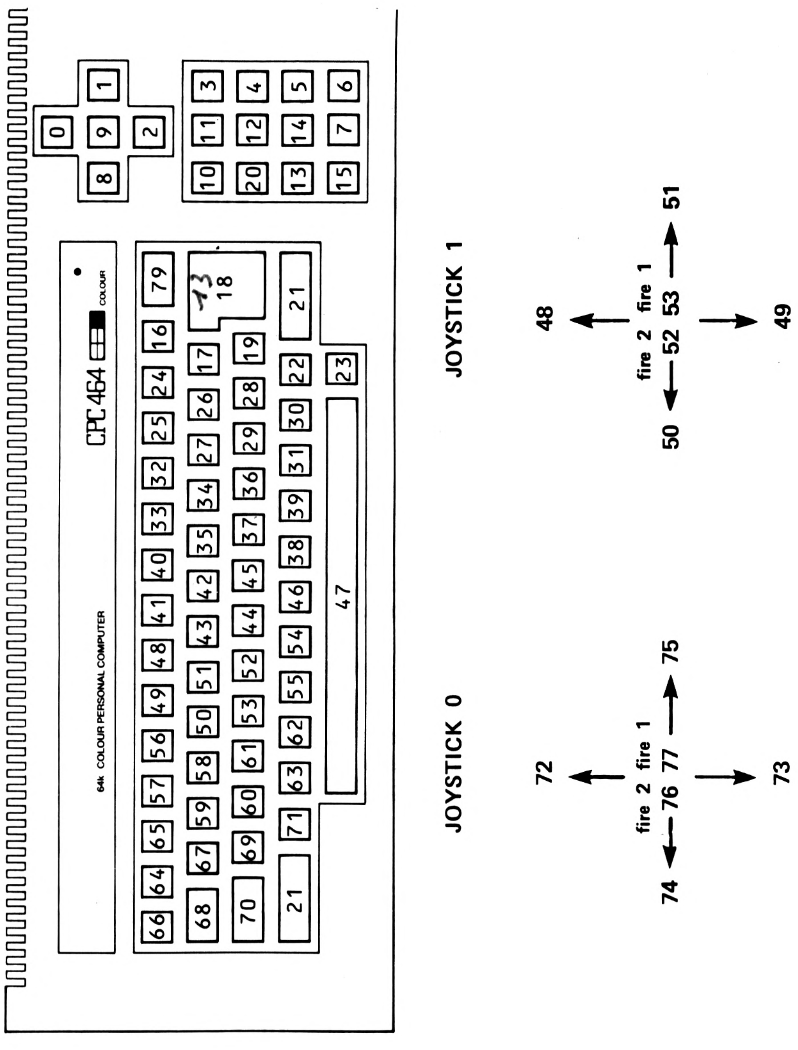
ASCII codes and CPC464 characters



exp char	value	ascii
128	0	30
129	1	31
130	2	32
131	3	33
132	4	34
133	5	35
134	6	36
135	7	37
136	8	38
137	9	39
138	.	2E
139	[enter]	0D
140	run"	52,55,4E,22,0D

Expansion characters, default locations and values

TOETSEN EN JOYSTICK NUMMERS



ASCII codes and CPC464 characters

AANHANGSEL IV

EEN INLEIDING VOOR GEVORDERDEN

De CPC464 is een kleuren Personal Computer die door het effectief gebruiken van de beste en beproefde technieken zo ontwikkeld werd, dat naast een bijzondere prijs-kwaliteit relatie ook veel ruimte voor uitbreidingsmogelijkheden wordt geboden en zowel voor beginners als voor gevorderden zeer interessant is.

De hardware en de ROM-software werden zo ontwikkeld, dat zowel de beginner als de gevorderde programmeur de mogelijkheid gegeven is, bestaande software aan te passen, nieuwe software te schrijven en de vele eigenschappen van de CPC464 te gebruiken.

DE HOOFDKENMERKEN ZIJN:

Z80 CPU

De meest verbreide microprocessor die wereldwijd in huis-computers gebruikt wordt, met de beste ondersteunde software basis - de CPC464 heeft de mogelijkheid om CP/M te gebruiken - De unieke onderbrekingsbehandeling maakt het mogelijk, dat de CPC464 de nieuwe BASIC instructies AFTER en EVERY gebruikt, evenals andere 'real time' functies in samenhang met geluid en tijdregistratie door de direkte beschikking over 4 timers.

64K RAM

Een royale hoeveelheid RAM die standaard ingebouwd is waarvan 42K de gebruiker ter beschikking staat. Dit dankzij de ROM overlappingstechniek, wanneer BASIC gebruikt wordt.

Beeldscherm

De CPC464 heeft 3 beeldscherm modi, inclusief 80 kolommen tekst, 27 kleuren en een hoog oplossend vermogen van 640 * 200 beeldpunten (PIXELS).

Toetsenbord

Een echt schrijfmachinetoetsenbord met cursortoetsen en een cijferblok, dat ook als functietoetsengroep gebruikt kan worden.

Cassetterecorder

Een datacassetterecorder is ingebouwd, zodat de gebruiker geen problemen heeft met de aansluiting, sterkteregeling, enz.. Schrijven is mogelijk met 1K baud of 2K baud, de leessnelheid stemt automatisch af op de software.

Basic

Een standaard BASIC die sneller en veelzijdiger is dan wat u van een BASIC zou verwachten; die met uitbreiding voor graphics en geluid als ook uitgebreide ondersteuning van de firmware uitgerust is.

Uitgebreide tekenset

Een 8 bits tekenset met inbegrip van symbolen en graphics-tekens, die grotendeels via het toetsenbord en met bijv. de CHR$(n) functie aangesproken kunnen worden.

Verbruikte tijd

Gesynchroniseerde onderbrekingen maken het mogelijk tijdmetingen te doen.

Gebruikers gedefinieerde toetsen

Tot en met 32 toetsen kan de gebruiker met ieder tot 32 strings definieren. De mogelijkheid om toetsen te definieren omvat ook de herhalingsfaktor. Een complete set van 255 tekens (de gehele ASCII tekenset en meer dan 100 andere) staat ter beschikking, die door de gebruiker herdefinieerd kan worden.

Subroutines

Vele assembler-subroutines kunnen via BASIC opgeroepen worden.

Beeldscherm modi

Er zijn drie beeldscherm modi:

a) Normaal

Mode 1: 40 kolommen * 25 regels, 4 tekstkleurmogelijkheden
        320 * 200 beeldpunten, ieder adresseerbaar in 4 kleuren

b) Veel kleuren modus

Mode 0: 20 kolommen * 25 regels, 16 tekstkleurmogelijkheden
160 * 200 beeldpunten, ieder adresseerbaar in 16 kleuren

c) Hoge resolutie

Mode 2: 80 kolommen * 25 regels, 2 tekstkleurmogelijkheden
640 * 200 beeldpunten, ieder adresseerbaar in 2 kleuren

Keuze der kleuren

(NB voor deze beschrijving geldt zwart (niet oplichten van het beeldscherm) als kleur)
Ongeacht welke beeldschermmodus wordt gebruikt, de rand kan ieder kleurenpaar aannemen, ofwel in twee kleuren knipperend of in 1 kleur. Het aantal ter beschikking staande INKten is van de beeldscherm modus afhankelijk. Iedere INKt kan twee kleuren knipperend of 1 enkele kleur aannemen. Het aantal te gebruiken INKten hangt af van de vooraf gedefinieerde beeldscherm modus. Het werkvlak (PAPER), de tekst-PEN en de graphics-PEN kunnen dan ieder een van de ter beschikking staande INKten krijgen.
De geschreven tekst kan ofwel transparant of dekkend zijn. Dat betekent dat de tekens al of niet in de achtergrondkleur op een anders gekleurd vlakje worden afgebeeld.

Windows - Vensters

U kunt tot acht tekstvensters kiezen, die met tekens beschreven kunnen worden en ook een graphicsvenster voor het uitbeelden van de graphics. De vensters worden geRESET, als u de modus voor het beeldscherm opnieuw kiest.
Opmerking: Als het tekstvenster het gehele beeldscherm omvat (normale toestand) dan zorgt de hardware voor het snel scrollen. Als een klein tekst venster gekozen wordt, dan geschiedt het scrollen doormiddel van de software en is dan langzamer.

Cursor

De cursor is buiten bedrijf indien de CPU geen invoer verwacht van het toetsenbord, dus het verschijnt pas wanneer hij om invoer vraagt. De cursor wordt als vierkant met inverse kleurenwaarde uitgebeeld.

Meerstemmig geluid

De geluiden die met de CPC464 gemaakt kunnen worden, worden door een standaard toongenerator van het type AY-3-8912 van General Instruments gerealiseerd. Het apparaat werkt met drie kanalen, waarin ieder onafhankelijk in toon en sterkte gestuurd kan worden. Witte ruis kan naar believen toegevoegd worden. De drie kanalen verschijnen links, rechts en in het midden (uitgang over de stereo aansluiting). De ingebouwde luidspreker geeft een gemengd mono geluid. De software biedt de mogelijkheid, de variatie van toon en geluidssterkte te sturen. De interne geluidssterkteregeling van de toon-generator wordt meestal niet gebruikt.

Printeraansluiting

Er is een standaard Centronics compatibele aansluiting voor parallel printers ingebouwd, waarbij het BUSY signaal voor HANDSHAKE operaties gebruikt wordt.

Uitbreidingen

Vele hardware interfaces zijn via JUMP blokken of andere inrichtingen aanspreekbaar, om de uitbouw van de software mogelijk te maken. Zo zullen van Schneider onder andere diskdrives, seriele interfaces met aangepaste software in ROM en vele andere worden geleverd.

Uitbreiding van de ROM

Alle ROMs bezetten de bovenste 16K van het geheugen (waar ook de BASIC zich bevindt) en via de firmware is het mogelijk tot 240 aanvullende 16K ROMs aan te spreken (daarbij wordt het basis apparaat met een adresdecoder uitgebreid, als onderdeel van de ROM uitbreidings-hardware).

Coordinaten

De tekst ORIGIN is de linker bovenhoek van het beeldscherm. De daadwerkelijke positie op het beeldscherm is van de beeldschermmodus afhankelijk.
De graphics ORIGIN begint links onder op het beeldscherm. Er wordt aangenomen, dat het beeldscherm voortdurend in high resolution (mode 2) werkt, hoewel de berekening van de INKten voor ieder beeldscherm-modus juist wordt doorgevoerd.
Let er wel op: In normale beeldschermmodus (mode 1) heeft ieder beeldpunt (pixel) TWEE horizontale adressen, een daarvan kan gebruikt worden. In de veel kleurenmodus (mode 0) heeft ieder beeldpunt VIER horizontale adressen, een der vier mag gebruikt worden om de positie te bepalen. De vertikale as heeft coordinaten van 0 tot 399, die door twee gedeeld en als integer de feitelijke positie van 0 tot 199 geven. Daardoor kunnen de proporties op het beeldscherm beter worden bepaald.

Overzicht

EEN KORTE SAMENVATTING VAN DE BELANGRIJKSTE EIGENSCHAPPEN VAN DE HARDWARE EN DE FIRMWARE VAN DE CPC464.

1) Hardware

1.1) BINNEN HET HUIS VAN DE CPC464

Computer, toetsenbord, cassetterecorder en luidspreker. RGB- en helderheids-uitgangen.

1.1.1) LSI CHIPS

Z80A processor met 4 MHZ klokfrequentie.
64K bytes met 64K * 1 dynamische RAM, opgefrist bij aansprken van het beeldschermgeheugen.
32K Bytes ROM, met BASIC en bedrijfssysteem (OS).

Een speciaal ontwikkelde logische schakeling bevat bijna alle logica, die niet in de LSI chips zit; bijvoorbeeld de timing, de kleuren-generatie en de DMA schakeling.
De 6845 CRT Controleeenheid levert de signalen voor het aftasten van de video RAM.

Let op

De ´mapping´ is gecompliceerd en verandert met de beeldscherm modi. De CRTC kan voor scrolling (zijdelings met 1/40 schermbreedte) en rolling (op en neer met 8 scan lijnen) door de software bestuurd worden.
Parameters in de CRTC bepalen het aantal regels, de rastering en de breedte en hoogten van de beeldschermrand.
De AY-3-8912 toongenerator van GI: 3 kanalen. Het geluid van de 3 kanalen wordt gelijkelijk gemengd en in mono over de ingebouwde luidspreker gestuurd, waarbij de geluidssterkte door een regelaar kan worden ingesteld. Er is ook een externe stereo - uitgang, waarbij geldt:
Links = kanaal A + 1/2 kanaal C ; Rechts = kanaal B + 1/2 kanaal C
Deze chip krijgt ook informatie van het toetsenbord of de joysticks.

Een 8255 parallelle I/O -chip bevat de interface van de bus naar de GI toongenerator. Het controleert ook het toetsenbord, de joystickpoort, cassetterecorder en andere apparaten.

1.2) BUITEN HET HUIS

De CPC464 heeft twee verschillende direkte aansluitingen voor monitoren. Beide monitoren hebben een 5 volt voedingsspanning voor de computer en zijn aan de netspanning in het land van verkoop aangepast. Er is ook de MP 1, een apparaat met een voedingsspanning en UHF modulator. Voor de aansluiting van HIFI apparatuur of een centronics printer heeft men een verbindingskabel nodig.

1.3) BEELDSCHERMTECHNIEK

Het beeldscherm werkt met een apart geheugen van 16K. De machine heeft 27 kleuren ter beschikking, die naar believen uitgezocht kunnen worden. Het aantal verschillende INKten op het palet hangt af van de beeldschermmodus. Meerdere INKten kunnen, indien nodig in dezelfde kleur worden toegewezen. De pixels op het beeldscherm zijn gedefinieerd als punten in een bepaalde kleur. (let er wel op dat de achtergrond, ofwel het papier (PAPER) een kleur uit de mogelijke reeks bevat.) De rand om het schrijfvlak heen (BORDER) kan volledig onafhankelijk een der 27 kleuren hebben.

MODE	INKten	vert.PUNTEN	horiz. PUNTEN	horiz. TEKENS
1 Normaal	4	200	320	40
2 Hoge resolutie	2	200	640	80
0 Meer kleuren	16	200	160	20

Op een monochrome monitor produceert de computer verschillende grijstinten. De helderheid van het beeld is als volgt:

KLEURENTABEL

Grijstint	Kleur	Grijstint	Kleur
0	zwart	14	pastel blauw
1	blauw	15	oranje
2	licht blauw	16	rose
3	rood	17	pastel magenta
4	magenta	18	licht groen
5	licht paars	19	zee groen
6	licht rood	20	licht cyaan
7	purper	21	citroen groen
8	licht magenta	22	pastel groen
9	groen	23	pastel cyaan
10	cyaan	24	licht geel
11	hemels blauw	25	pastel geel
12	geel	26	helder wit
13	grijs		

1.4) GEHEUGEN INDELING

De 64K RAM is als volgt ingedeeld:

Let er wel op, dat een deel van de ROM het scherm RAM (beeldscherm-geheugen) overlapt en daarbij het groots mogelijke gebied voor de gebruikersRAM tijdens BASIC operaties vrij laat.

```
                  0000H
ROM SECTION 0 <
                  3FFFH

                  4000H
                  7FFFH
                  8000H
                  BFFFH

                  C000H
ROM SECTION 1 <             > RAM [scherm 3]
                  FFFFH
```

Als zowel RAM als ook ROM op het gelijke adres liggen, dan wordt bij het lezen de ROM en bij het schrijven de RAM aangesproken. Ieder der beide ROM-bereiken kan uitgeschakeld worden, om de RAM op deze gelijke adressen te kunnen uitlezen.

1.5) AANVULLINGEN

1.5.1) ROMs

Aanvullende ROMs kunnen in de plaats van iedere ingebouwde ROM gezet worden. De adressenkeuze en de 'bank selection' logica bevinden zich in een module, dat aan de uitbreidingsbus aangesloten kan worden. Alle signalen die daar nodig zijn, worden naar de uitbreidingsbus gestuurd.

1.5.2) RAM UITBREIDING

Aanvullende RAM kan op iedere plaats, in de plaats van de ingebouwde RAM bijgevoegd worden. De adressenkeuze en de 'bank selection' logica zijn in een module dat aan de uitbreidingsbus aangesloten wordt ondergebracht. Alle signalen, die daar nodig zijn, worden naar de uitbreidingsbus gestuurd. Deze geheugenplaats kan alleen gelezen worden en om I/O operaties mogelijk te maken is nog een speciale toevoeging vereist, die de computer toestaat, naar deze RAM-toevoeging te schrijven.

1.5.3) TOEVOEGING IN- UITVOER (I/O)

De meeste adressen voor de I/O aansluitingen zijn door de computer gereserveerd. In geen geval mogen de adressen beneden **7Fxx** gebruikt worden. De volgende adressen kunnen voor externe hardware gebruikt worden.

F8xx, F9xx, FAxx, FBxx

Rand-apparaten aan de uitbreidingsbus moeten de adressen op **A0** tot **A7** decoderen; waarbij adres **A10** 'low' is. De I/O kanalen aan de uitbreidingsbus in het adressengebied **F800** tot **FBFF** zijn als volgt gereserveerd:

Adressen A0-A7

00-7B	*** niet gebruiken ***
7C-7F	**gereserveerd voor disk-drive interface**
80-BB	*** niet gebruiken ***
BC-BF	**gereserveerd voor toekomstig gebruik**
C0-DB	*** niet gebruiken ***
DC-DF	**gereserveerd voor communicatie interfaces**
E0-FF	**staan de gebruiker voor rand-apparaten ter beschikking**

Let er wel op, dat de Z80 instructies gebruikt moeten worden, die het **B** register in de bovenste helft van de adressenbus (**A15-A8**) zetten.

2) Toetsenbord

Een volledige RESET geschiedt door het gelijktijdig indrukken van **[CRTL] [SHIFT] [ESC]**. Toetsen die tekens of cursor-bewegingen produceren, hebben een door de firmware gestuurde herhalingsfunctie, uitgezonderd alle toetsen in het afzonderlijke cijferblok.

[ESC] onderbreekt de uitvoering van een programma. Wordt een tweede **[ESC]** gegeven, dan wordt de uitvoering gestopt. Drukt men daarentegen op een andere toets, dan wordt de uitvoering voortgezet.

[CAPS LOCK] is een omschakelingsfunctie, die via de caps lock toets gestuurd wordt. Shift lock is een omschakelingsfunctie, die bekrachtigd wordt, als men **[CTRL][CAPS LOCK]** gelijktijdig indrukt.

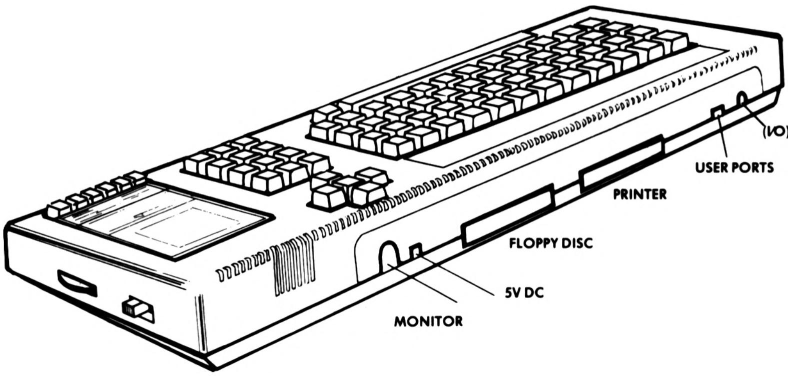
De copy-cursor wordt van de invoer-cursor gescheiden, als men een cursor-toets gelijktijdig met de **[SHIFT]** toets indrukt. Men krijgt het teken, dat onder de copy-cursor staat in de invoerregels als men de **[COPY]** toets indrukt.

De cursor toetsen maken het EDITten van de invoerbuffer mogelijk, die zich over meerdere beeldschermregels uitstrekken kan. Met deze toetsen kan de start van de toetsenbordinvoer op iedere beeldschermpositie gebracht worden, voordat een invoer over de toetsen volgt. Zodra de toetsenbordinvoer ontvangen wordt, is de beeldschermpositie vastgelegd. De nieuwe invoertekst overschrijft een voorhanden zijnde inhoud op deze beeldschermpositie.

[DEL] wist achteruit   **[CLR]** wist vooruit.

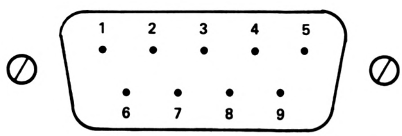
AANHANGSEL V:

Aansluitingen aan de achterzijde van de CPC464



PADDLE PORT CONNECTOR (9 PIN D)

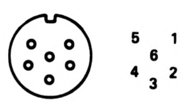
VIEWED FROM REAR



PIN 1	UP	PIN 6	FIRE 2
PIN 2	DOWN	PIN 7	FIRE 1
PIN 3	LEFT	PIN 8	COMMON
PIN 4	RIGHT	PIN 9	COM 2
PIN 5	SPARE		

VIDEO OUTPUT CONNECTOR (6 PIN DIN)

VIEWED FROM REAR

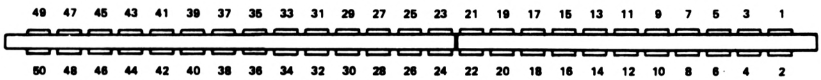


PIN 1	RED	PIN 4	SYNC
PIN 2	GREEN	PIN 5	GND
PIN 3	BLUE	PIN 6	LUM

The user interface and expansion connectors

EXPANSION PORT 50 WAY 0.1 EDGE CONNECTOR

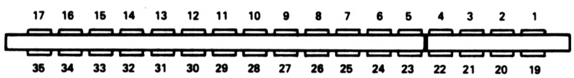
VIEWED FROM REAR



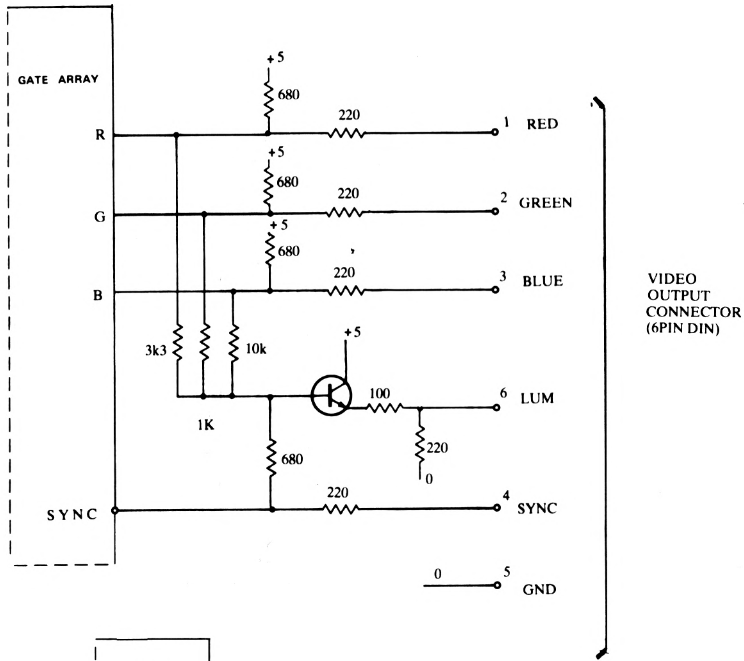
PIN 1	SOUND	PIN 18	A0	PIN 35	INT
PIN 2	GND	PIN 19	D7	PIN 36	NMI
PIN 3	A15	PIN 20	D6	PIN 37	BUSRD
PIN 4	A14	PIN 21	D5	PIN 38	BUSAK
PIN 5	A13	PIN 22	D4	PIN 39	READY
PIN 6	A12	PIN 23	D3	PIN 40	BUS RESET
PIN 7	A11	PIN 24	D2	PIN 41	RESET
PIN 8	A10	PIN 25	D1	PIN 42	ROMEN
PIN 9	A9	PIN 26	D0	PIN 43	ROMDIS
PIN 10	A8	PIN 27	+ 5v	PIN 44	RAMRD
PIN 11	A7	PIN 28	MREQ	PIN 45	RAMDIS
PIN 12	A6	PIN 29	M1	PIN 46	CURSOR
PIN 13	A5	PIN 30	RFSH	PIN 47	L. PEN
PIN 14	A4	PIN 31	IORQ	PIN 48	EXP
PIN 15	A3	PIN 32	RD	PIN 49	GND
PIN 16	A2	PIN 33	WR	PIN 50	ϕ
PIN 17	A1	PIN 34	HALT		

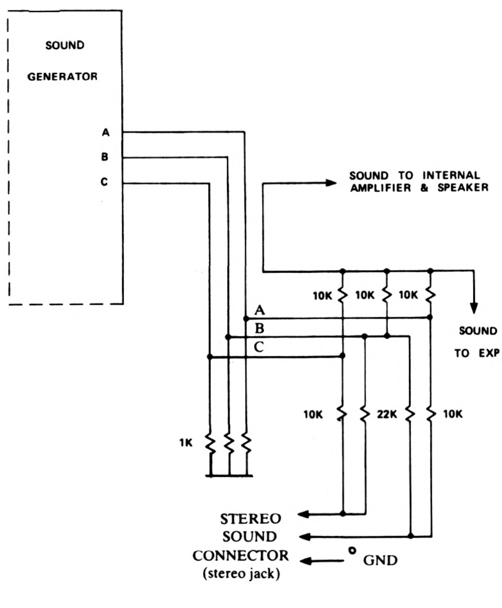
PRINTER PORT 34 WAY 0.1 EDGE CONNECTOR

VIEWED FROM REAR



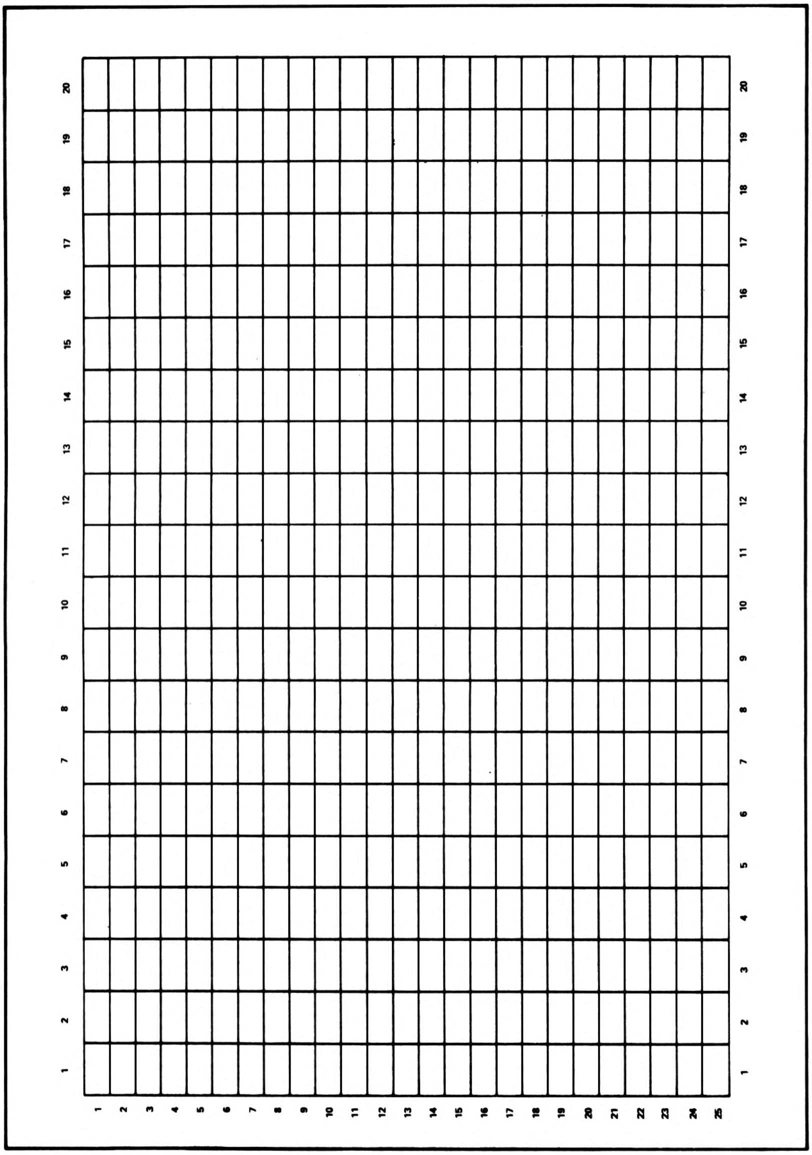
PIN 1	STROBE	PIN 19	GND
PIN 2	D0	PIN 20	GND
PIN 3	D1	PIN 21	GND
PIN 4	D2	PIN 22	GND
PIN 5	D3	PIN 23	GND
PIN 6	D4	PIN 24	GND
PIN 7	D5	PIN 25	GND
PIN 8	D6	PIN 26	GND
PIN 9	D7	PIN 27	GND
PIN 11	BUSY	PIN 28	GND
PIN 14	GND	PIN 33	GND
PIN 16	GND	All other pins NC	





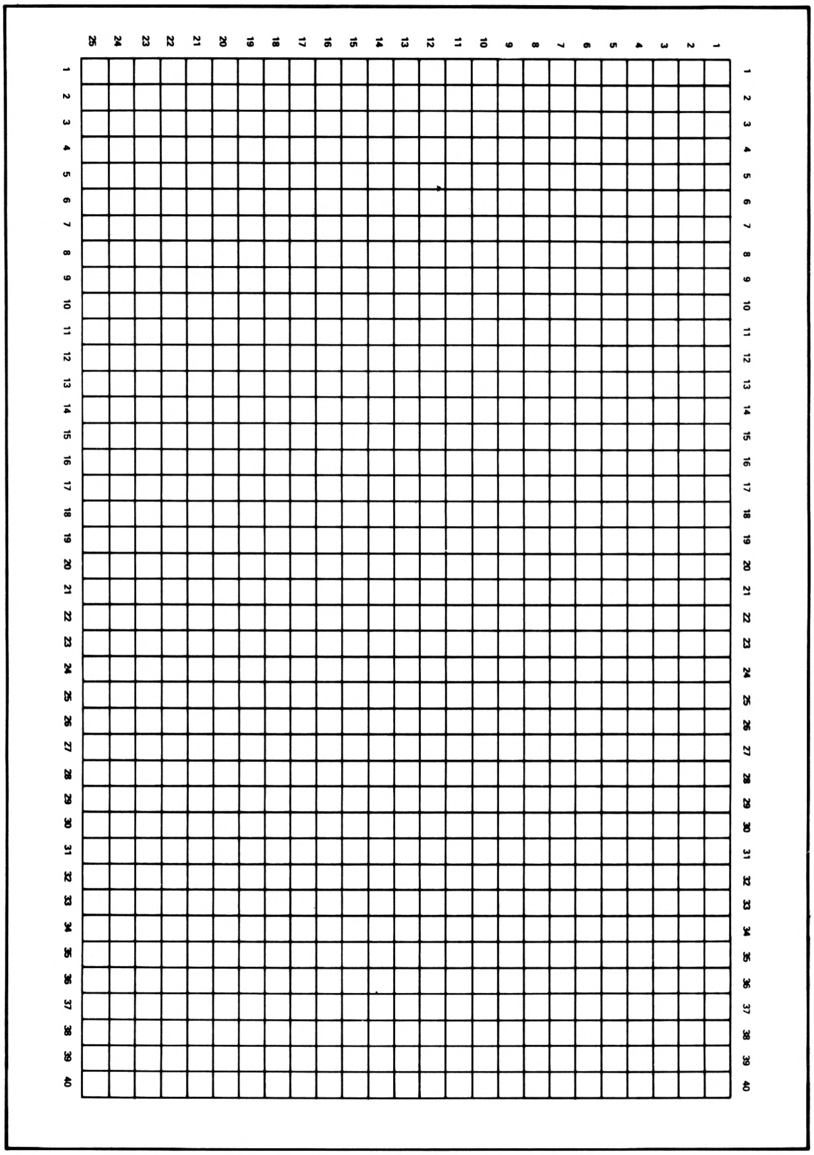
The user interface and expansion connectors

TEXT and WINDOW planner
MODE 0   20 columns

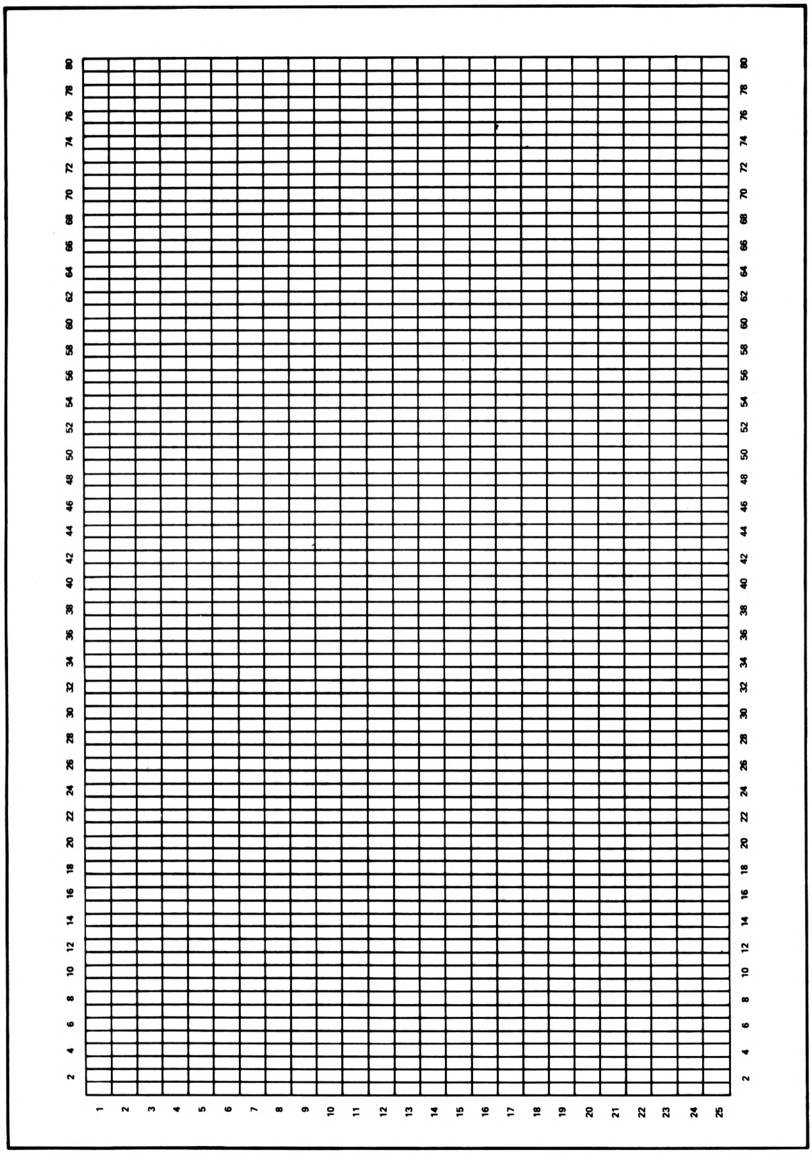


Graphics and text planners

Text and WINDOW planner
Mode 1   40 columns



Text and WINDOW planner
Mode 2 80 Columns



Graphics and text planners

AANHANGSEL VII

NOTEN en TOON HOOGTEN

De volgende tabellen beschrijven de aanbevolen toonperioden voor de noten van de acht octaven.

De werkelijk geproduceerde frequentie is niet exact de gewenste frequentie, daar de periode een integer moet zijn. De relatieve fout is de verhouding van het verschil tussen de verlangde en de bereikte frequentie, volgens (verlangde - bereikte)/verlangde).

NOOT FREQUENTIE PERIODE RELATIEVE FOUT

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	32.703	3822	-0.007%	
C#	34.648	3608	+0.007%	
D	36.708	3405	-0.007%	
D#	38.891	3214	-0.004%	
E	41.203	3034	+0.009%	
F	43.654	2863	-0.016%	Octave -3
F#	46.249	2703	+0.009%	
G	48.999	2551	-0.002%	
G#	51.913	2408	+0.005%	
A	55.000	2273	+0.012%	
A#	58.270	2145	-0.008%	
B	61.735	2025	+0.011%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	65.406	1911	-0.007%	
C#	69.296	1804	+0.007%	
D	73.416	1703	+0.022%	
D#	77.782	1607	-0.004%	
E	82.407	1517	+0.009%	
F	87.307	1432	+0.019%	Octave -2
F#	92.499	1351	-0.028%	
G	97.999	1276	+0.037%	
G#	103.826	1204	+0.005%	
A	110.000	1136	-0.032%	
A#	116.541	1073	+0.039%	
B	123.471	1012	-0.038%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	130.813	956	+0.046%	
C#	138.591	902	+0.007%	
D	146.832	851	-0.037%	
D#	155.564	804	+0.058%	
E	164.814	758	-0.057%	
F	174.614	716	+0.019%	**Octave -1**
F#	184.997	676	+0.046%	
G	195.998	638	+0.037%	
G#	207.652	602	+0.005%	
A	220.000	568	-0.032%	
A#	233.082	536	-0.055%	
B	246.942	506	-0.038%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	261.626	478	+0.046%	Middle C
C#	277.183	451	+0.007%	
D	293.665	426	+0.081%	
D#	311.127	402	+0.058%	
E	329.628	379	-0.057%	
F	349.228	358	+0.019%	**Octave 0**
F#	369.994	338	+0.046%	
G	391.995	319	+0.037%	
G#	415.305	301	+0.005%	
A	440.000	284	-0.032%	International A
A#	466.164	268	-0.055%	
B	493.883	253	-0.038%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	523.251	239	+0.046%	
C#	554.365	225	-0.215%	
D	587.330	213	+0.081%	
D#	622.254	201	+0.058%	
E	659.255	190	+0.206%	
F	698.457	179	+0.019%	**Octave 1**
F#	739.989	169	+0.046%	
G	783.991	159	-0.277%	
G#	830.609	150	-0.328%	
A	880.000	142	-0.032%	
A#	932.328	134	-0.055%	
B	987.767	127	+0.356%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	1046.502	119	-0.374%	
C#	1108.731	113	+0.229%	
D	1174.659	106	-0.390%	
D#	1244.508	100	-0.441%	
E	1318.510	95	+0.206%	
F	1396.913	89	-0.543%	Octave 2
F#	1479.978	84	-0.548%	
G	1567.982	80	+0.350%	
G#	1661.219	75	-0.328%	
A	1760.000	71	-0.032%	
A#	1864.655	67	-0.055%	
B	1975.533	63	-0.435%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	2093.004	60	+0.462%	
C#	2217.461	56	-0.662%	
D	2349.318	53	-0.390%	
D#	2489.016	50	-0.441%	
E	2637.021	47	-0.855%	
F	2793.826	45	+0.574%	Octave 3
F#	2959.955	42	-0.548%	
G	3135.963	40	+0.350%	
G#	3322.438	38	+0.992%	
A	3520.000	36	+1.357%	
A#	3729.310	34	+1.417%	
B	3951.066	32	+1.134%	

NOTE	FREQUENCY	PERIOD	RELATIVE ERROR	
C	4186.009	30	+0.462%	
C#	4434.922	28	-0.662%	
D	4698.636	27	+1.469%	
D#	4978.032	25	-0.441%	
E	5274.041	24	+1.246%	
F	5587.652	22	-1.685%	Octave 4
F#	5919.911	21	-0.548%	
G	6271.927	20	+0.350%	
G#	6644.875	19	+0.992%	
A	7040.000	18	+1.357%	
A#	7458.621	17	+1.417%	
B	7902.133	16	+1.134%	

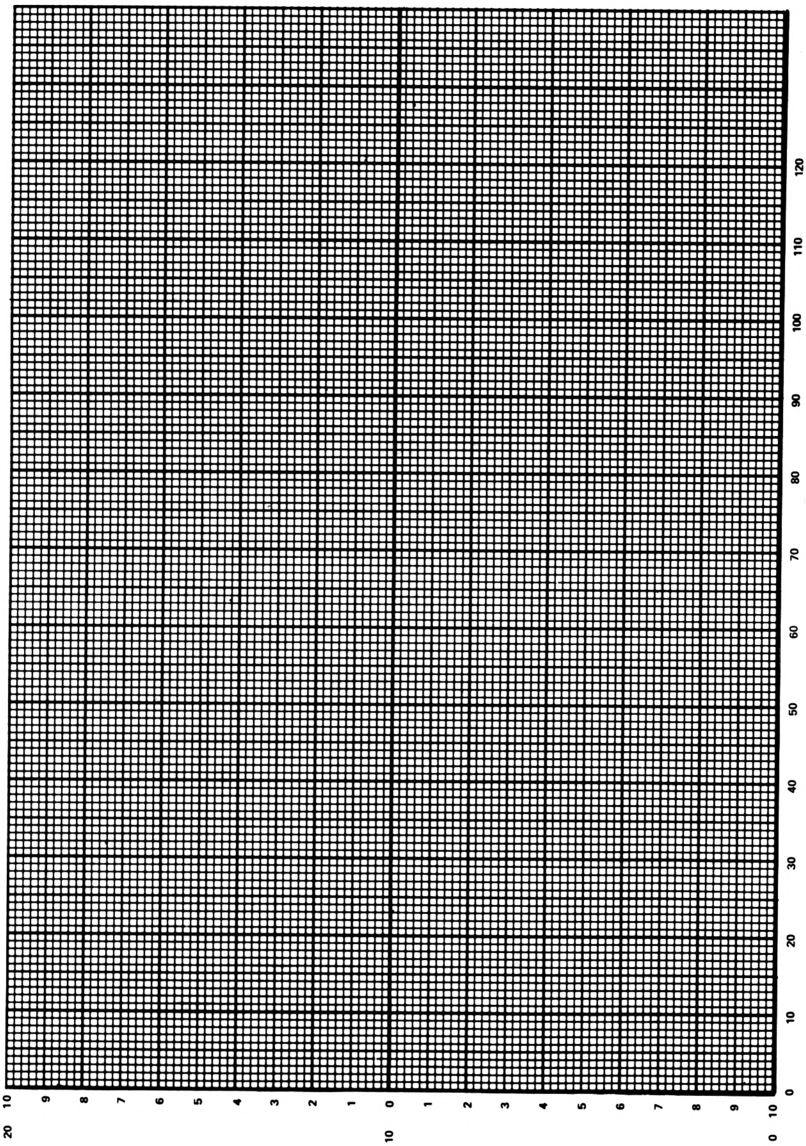
These values are all calculated from International A as follows :

FREQUENCY = 440*(2↑(OCTAVE+((N−10)/12)))
PERIOD = ROUND(125000/FREQUENCY)

where N is 1 for C, 2 for C#, 3 for D etc.

Music planner

Sound envelope/Music planner sheet



Music planner

AANHANGSEL VIII:

FOUTCODES en GERESERVEERDE WOORDEN

Foutnummers en foutmeldingen

Als BASIC een programma instructie, woord of variabele aantreft dat het niet bewerken kan, stopt het en geeft een fout-melding. De foutmelding beschrijft de oorzaak. Indien het alleen om een typefout gaat, schakelt BASIC in de EDIT mode en geeft de te corrigeren regel op het scherm.

De bekendste fout, die door onnauwkeurig intypen veroorzaakt wordt, is de **Syntax error** (nummer 2). Indien de fout in de programmaloop is opgetreden, geeft BASIC de regel opnieuw, zodat deze direkt gecorrigeerd kan worden. In de direkte invoer mode wordt alleen de foutmelding gegeven, daar aangenomen wordt dat de laatst ingetikte regel de fout bevat.

Als de ON ERROR GOTO instructie aan het programmabegin gegeven werd, wordt bij het optreden van een fout naar de aangegeven regel gesprongen. In het volgende voorbeeld wordt er naar regel 1000 gesprongen indien er een fout optreedt.

```
 10 ON ERROR GOTO 1000

    programma

 1000 PRINT CHR$(7):MODE 2:INK 1,0:INK 0,13:CLS:LIST
```

Waarop de CPC464 piept, het beeldscherm wist, op een passende kleuren-combinatie voor de 80 koloms modus schakelt en het programma LIST, om te laten onderzoeken wat er aan de hand is. Indien een **Syntax error** de foutoorzaak was, verschijnt de te corrigeren regel aan het einde van de LIST, en BASIC wacht in de lijn EDIT mode op een correctie, ofschoon de foutmelding onderdrukt werd.

Vergeet niet, het programma voor regel 1000 te beeindigen (END), als u de uitvoer op het beeldscherm wilt behouden.

BASIC produceert voor geldige invoer geen foutmelding, zodat indien later toch een fout optreedt, deze op een programmeerfout terug te voeren is, waarbij de fout melding het meest helpt, om de oorzaak op te sporen. Het meest leert men uit zijn eigen fouten, de CPC464 is een geduldige leraar: u geeft het eerder op dan de CPC464

Alle fouten die BASIC produceren kan, zijn hierna met hun foutnummers afgedrukt. Waarbij de foutmelding in een korte beschrijving is weergegeven.

1. Unexpected NEXT - Een onverwachte NEXT

Een NEXT instructie werd gevonden, maar kwam geen FOR bewerking tegen, of de stuurvariabele in de NEXT instructie past niet bij de FOR.

2. Syntax Error - Syntax fout

BASIC herkent de regel niet, om dat er iets in zit dat het niet begrijpt.

3. Unexpected RETURN - onverwachte RETURN

Een RETURN instructie werd buiten een subroutine gevonden.

4. DATA exhausted - data ten einde.

Een READ instructie werd gegeven zonder dat de dataregels nog gegevens bevatten.

5. Improper argument - ongeldig argument

Deze melding heeft meerdere betekenissen. De waarde van een argument van een functie of een instructie parameter is fout.

6. Overflow - overloop

Het resultaat van een berekening heeft het toelaatbare overschreden. Het kan een decimaal getal overloop zijn in welk geval de bereking de grootst toegelaten waarde: **1.7E-38** overschreden heeft. Dit kan ook het geval zijn wanneer getracht werd een decimaal getal om te zetten in een 16 bits integer.

7. Memory full - geheugen is vol

Het programma of zijn variabelen zijn te groot, of het programma is te ingewikkeld (met elkaar verbonden GOSUBs, WHILEs en FORs). Een MEMORY instructie kan deze melding veroorzaken, indien de BASIC ter beschikking staande geheugenruimte te klein is of een buiten de mogelijkheden vallend ruimte werd aangegeven. Let er wel op dat een open cassette file via een buffer loopt, zodat de beschikbare geheugen-ruimte kleiner wordt.

8. Line does not exist - regel is niet voorhanden.

De aangegeven regel bestaat niet.

9. Subcript out of range - subscript buiten het geoorloofde bereik.

Een subscript (index) in een matrix is te groot of te klein.

10. Array already dimensioned - matrix is reeds gedefinieerd.

Een der matrixen in een DIM instructie werd al gedefinieerd.

11. Division by zero - delen door nul

Kan bij delen door natuurlijke- of integer getallen, bij integer-modulus of bij exponenten.

12. Invalid direct command - foutieve direkte instructie.

De laatst geprobeerde instructie is in direkte modus niet geoorloofd.

13. Type mismatch - uitdrukking fout.

Deze fout treedt op als een getal gebruikt wordt in plaats van een string, of omgekeerd, of er werd in een READ of INPUT een getal van een fout type ingevoerd.

14. String space full - string gebied vol.

Er werden teveel strings gemaakt zodat er geen plaats meer is, zelfs niet na een ´garbage collection´.

15. String too long - string te lang.

Een string kan maximaal 255 tekens lang zijn. Kan optreden wanneer er meerdere strings aan elkaar gevoegd worden.

16. String expression too complex - stringuitdrukking te complex.

Stringuitdrukkingen kunnen vele tijdelijke stringwaarden voortbrengen. Wanneer het aantal een redelijke grens overschrijdt, geeft BASIC deze melding.

17. Cannot CONTinue - CONT werkt niet.

Treedt op wanneer het programma niet opnieuw gestart kan worden met CONT wanneer het geRUNd heeft. Bij een STOP werkt het normaal wel.

18. Unknown user function - onbekende gebruikersfunctie.

De aangeroepen functie werd niet door DEF FN gedefinieerd.

19. RESUME missing - RESUME ontbreekt.

Het programmaeinde werd tijdens een foutbehandelingsroutine bereikt, bijv. in een ON ERROR GOTO routine.

20. Unexpected RESUME - onverwachte RESUME

RESUME is alleen toegestaan bij een foutbehandeling, zoals een ON ERROR GOTO routine.

21. Direct command found - direkte instructie gevonden.

Bij het laden van een programma van cassette werd een regel zonder nummer gevonden.

22. Operand missing - operand is er niet.

BASIC heeft een onvolledige instructie gevonden.

23. Line too long - regel te lang

Een regel wordt voor de interne BASIC te groot.

24. EOF met - EOF aangetroffen.

Er werd geprobeerd een bestand van cassette te lezen, voorbij het einde van het bestand.

25. File type error - data type onjuist

De ingelezen cassettefile is van een ongeoorloofd type. OPENIN is alleen voor ASCII-tekst geoorloofd. LOAD, RUN, enz. zijn alleen voor files geoorloofd, die met SAVE weggeschreven werden.

26. NEXT missing - NEXT ontbreekt.

Een FOR werd gegeven, maar een passende NEXT is niet gevonden.

27. File already open - file is reeds open.

Een OPENIN of OPENOUT instructie werd gegeven, ofschoon de vorige file nog niet gesloten is.

28. Unknown command - onbekende instructie.

Een voor BASIC onbekende externe instructie.

29. WEND missing - WEND ontbreekt.

Een WHILE instructie werd gegeven en een passende WEND kan niet gevonden worden.

30. Unexpected WEND - onverwachte WEND.

Een WEND werd buiten een WHILE bewerking gevonden, of een WEND past niet bij de actuele WHILE bewerking.

BASIC WOORDEN

De volgende BASIC woorden zijn gereserveerd en mogen niet als variabelenamen gebruikt worden.

ABS, AFTER, AND, ASC, ATN, AUTO

BIN$, BORDER

CALL, CAT, CHAIN, CHR$, CINT, CLEAR, CLG, CLOSEIN, CLOSEOUT, CLS, CONT, COS, CREAL

DATA, DEF, DEFINT, DEFREAL, DEFSTR, DEG, DELETE, DI, DIM, DRAW, DRAWR

EDIT, EI, ELSE, END, ENT, ENV, EOF, ERASE, ERL, ERR, ERROR, EVERY, EXP

FIX, FN, FOR, FRE

GOSUB, GOTO

HEX$, HIMEM

IF, INK, INKEY, INKEY$, INP, INPUT, INSTR, INT

JOY

KEY

LEFT$, LEN, LET, LINE, LIST, LOAD, LOCATE, LOG, LOG10, LOWER$

MAX, MEMORY, MERGE, MID$, MIN, MOD, MODE, MOVE, MOVER

NEXT, NEW, NOT

ON, ON BREAK, ON ERROR GOTO, ON SQ, OPENIN, OPENOUT, OR, ORIGIN, OUT

PAPER, PEEK, PEN, PI, PLOT, PLOTR, POKE, POS, PRINT

RAD, RANDOMIZE, READ, RELEASE, REM, REMAIN, RENUM, RESTORE, RESUME, RETURN, RIGHT$, RND, ROUND, RUN

SAVE, SGN, SIN, SOUND, SPACE$, SPC, SPEED, SQ, SQR, STEP, STOP, STR$, STRING$, SWAP, SYMBOL

TAB, TAG, TAGOFF, TAN, TEST, TESTR, THEN, TIME, TO, TROFF, TRON

UNT, UPPER$, USING

VAL, VPOS

WAIT, WEND, WHILE, WIDTH, WINDOW, WRITE

XOR, XPOS

YPOS

ZONE

BEGRIPPEN EN HUN VERKLARINGEN

Een aantal veel gebruikte begrippen uit de wereld van de computertechniek en de verklaring voor de CPC464 gebruiker.

ACCUMULATOR

Een geheugenlokatie in het microprocessor circuit in het hart van de microcomputer, dat tijdelijk gegevens opslaat terwijl het wordt verwerkt. Wordt veel en uitgebreid in machinecode gebruikt - BASIC gebruikers zullen niet eens van het bestaan afweten.

ACOUSTIC COUPLER

Ook bekent als een acoustische modem. Een electronische toevoeging dat het verbinden van een telefoon met een computer verzorgd en zodoende de computer toestaat te communiceren via de normale telefoon Op deze manier kan de computer in contact treden met openbare informatiesystemen zoals VIDITEL, Bulletin boards enz. en andere gebruikers om gegevens uit te wisselen, informatie te verzamelen enz..

ADRES - ADDRESS

Het nummer in een instructie ter aanduiding van een cel in het geheugen van de computer. Door zijn adres kan een bepaalde geheugen-locatie uitgekozen worden, zodat de inhoud uitgelezen of beschreven kan worden.

ADVENTURE GAMES - AVONTUREN SPEL

Voor sommigen het beste voor anderen het slechtste. Een computer-spel met veel tekst, met toevalsgebeurtenissen waarin men zijn weg moet zoeken in een doolhof of labyrinth.

ALGORITHME

Een volgorde van logische en rekenkundige stappen om een specifiek probleem op te lossen.

ALPHANUMERIEK

Samentrekking van alfabetisch en numeriek, waarmee aangegeven wordt, dat gegevens bestaan uit cijfers en alfabetische tekens.

ALU

Afkorting voor Arithmetic Logic Unit, het gedeelte van de CPU, microprocessor, dat de logische en rekenkundige bewerkingen uitvoert.

A/D Analoog

Het omzetten van gelijkmatig verlopende signalen in een aantal stappen. Computers werken digitaal, zodat de in de natuurlijke wereld voorkomende gebeurtenissen eerst in digitale eenheden omgevormd moeten worden, voordat de computer daar iets mee kan beginnen.

ANIMATIE

Tekenfilms zijn de bekendste vorm van animatie. Bij computer-animatie worden grafische uitbeeldingen zo gegeven dat het een schijn van echte bewegingen geeft.

APPLICATIE PROGRAMMA
Een programma dat ontworpen is voor een specifieke toepassing.

ARCADE GAMES
Computerspel met veel aktie, reflexen, ruimtemonsters en allerlei onplezierige manieren om het leven te verliezen. Weinig educatieve waarde.

ARCHITECTUUR
Algemene opbouw in relatie tot de databus, randapparatuur en CPU van een microcomputer. Niet een thema om hier te behandelen.

ARGUMENT
Een onafhankelijke variabele, bijv in de uitdrukking x+y=z zijn x en y de argumenten.

ARRAY
Een 2 dimensionale matrix (raster) waarin gegevens zijn opgeslagen door adressering met de ´horizontale´ en ´verticale´ coordinaten.

ASCII
American Standard Code for Information Interchange. Een veel gebruikte wijze voor het weergeven van getallen, letters en andere symbolen die via een toetsenbord ingevoerd kunnen worden of door andere instructies aangeroepen kunnen worden. De CPC464 codes staan in aanhangsel III.

ASSEMBLER
Het programma dat symbolische code vertaalt in machine-code.

BAR CODE
Een code die bestaat uit een aantal inktlijnen en die afgedrukt kan worden op een etiket of een label. Door optische lezing is de computer in staat dit te vertalen en te verwerken.

BASE
De basis in getallensystemen. Het binaire systeem heeft basis 2, het decimale systeem heeft basis 10 en het hexadecimale systeem heeft de basis 16. Zie aanhangsel II voor verdere informatie.

BASIC
Beginner´s All-purpose Symbolic Instruction Code. Als interpretatieve programmeertaal, gebruikt op bijna alle huiscomputers, was BASIC speciaal ontworpen om snel en makkelijk te leren of te begrijpen. Programma´s kunnen makkelijk aan elkaar gelijmd worden en ook op elk moment uitgetest worden. Dit in tegenstelling tot gecompileerde talen.

BAUD
Een bit per seconde: de eenheid waarmee de overdrachtssnelheid van serieele communicatie apparatuur wordt aangegeven.

BCD
Binary Coded Decimal. Een codeer systeem voor decimale getallen waarin ieder cijfer met vier bits wordt bepaald.

BENCHMARK
Standaard opgaven voor computers, waarmee zij vergeleken kunnen worden op hun snelheid, efficienty en accuratesse.

BINAIR
(zie Base) Het getalstelsel met basis 2, waarin alle getallen gevormd worden door de binaire cijfers 0 en 1.

BINAIR GETAL
Een getal weergegeven in binaire notatie. Aangeduidt in CPC464 programma's door het voorvoegsel **&X.**

BIT
Afkorting voor BInary digiT. Een binair cijfer kan in het binaire getalstelsel alleen maar de waarde 0 of 1 aannemen.

BIT SIGNIFICANT
Ieder van de acht bits van een byte heeft een eigen betekenis (0 of 1). Voor het uitwerken moet ieder bit getest worden.

BOOTING OF BOOTSTRAPPING
Programma's of Operating Systems laden niet uit zichzelf, zij moeten gestart worden door een kleine routine in ROM (gewoonlijk). Deze routine initialiseert het laadproces op een bepaalde geheugenplaats.

BOOLEAANSE ALGEBRA
Een opdracht met logische relaties, waaruit maar twee antwoorden kunnen voortvloeien: TRUE of FALSE; 0 of 1.

BUFFER
Een (tussen)geheugen, dat overwegend gegevens opneemt die van het ene gedeelte van het systeem naar het andere worden overgedragen; bijv van de cassetterecorder naar de microprocessor en hoofd-RAM. Een buffer controleert en reguleert de gegevensstroom tussen verschillende apparaten, die met verschillende snelheden werken zoals een modem of een printer.

BUG
Een probleem dat steeds verschillend kan optreden, van bijv. bij het onjuist gebruiken van een programma (bijv. de kleur van het scherm verandert, wanneer u vier toetsen tegelijk indrukt) tot een reeks van fouten dat geheel en onherstelbaar het programma verstoord en het geheugen geheel verschoond van gegevens.

BUS
Een groep van verbindigen die zowel voor verbinding binnen de computer zorgen als naar buiten de computer met informatie over de toestand van de CPU, de RAM of andere hardware eigenschappen. De CPC464 bus is op de grotere van de twee printplaatconnectoren aan de achterzijde van de computer aangebracht.

BYTE
Een groep van acht bits, welke de kleinste adresseerbare eenheid in het geheugen is.

CAD
Computer Aided Design. Normalerwijze een samenspel van reken en graphics kwaliteiten, gebruikt als electronische tekentafel, alhoewel iedere berekening op een computer uitgevoerd onder de kop CAD wordt geschaard.

CAE
Computer Aided Education. Het gebruik van de computer in het onderwijs. CAI (Computer Aided Instruction: instructie m.b.v. de computer) en CAL (Computer Aided Learning: leren m.b.v. de computer) zijn twee aspecten van CAE.

CARTRIDGE
Verpakte geheugen eenheid met een geintegreerde schakeling die software inhoudt en direkt in een speciaal daarvoor aanwezige konnektor geplaatst kan worden. Het programma is vrijwel direkt beschikbaar voor gebruik.

CASSETTE
Buiten de bekende magneetband verstaat men hieronder alles wat verpakt komt, ROM software enz..

CHARACTER
Een letter, cijfer of een grafisch symbool, dat in de computer ingevoerd kan worden. Zie aanhangsel III.

CHARACTER CELL
Puntenmatrix op het beeldscherm, waarin tekens in verscheidene kleuren zichtbaar worden gemaakt.

CHARACTER SET
Alle letters, cijfers of grafische symbolen die in een computer of printer ter beschikking staan. Hetgeen niet wil zeggen dat als ze op de computer voorhanden zijn, dat ook zo voor de printer geldt.

CHARACTER STRING
Meerdere tekens die als eenheid bewaard kunnen worden of behandeld. Bijv. een woord of een rij van woorden.

CHIP
Een misleidend, jedoch in de spreektaal veel gebruikt begrip voor iedere geintegreerde schakeling. De chip is in werkelijkheid een dunne schijf speciaal vervaardigde silicium, waarop de schakeling is ondergebracht.

COMPILER
Een complex programma dat andere in hogere programmeertalen geschreven programma´s, zoals BASIC, omzet in de direkte instructie-code van de microprocessor. Hierdoor wordt een grotere verwerkings-snelheid mogelijk.

CP/M
Digital Research´s de facto standaard voor 8-bit computer disk operating systems.

CPU
Central Processing Unit. Het hart van ieder computersysteem, waarin de instructies geinterpreteerd worden en vervolgens uitgevoerd. In een microcomputer is de CPU de microprocessor zelf.

CURSOR
Een beweegbaar merkteken, dat aangeeft waar het volgende teken op het scherm verschijnt.

CURSOR CONTROLE TOETSEN
Toetsen, waarop pijlen zijn afgebeeld, die de cursor over het scherm doen bewegen. Worden veel in spellen gebruikt voor controle over bewegingen.

DAISY-WHEEL PRINTER
Een printer met een hoge kwaliteit letters en cijfers. Geprinte tekens verkrijgt men door een vorm van het teken tegen een inktlint aan te drukken.

DATABASE
Een matrix van ieder type gegevens in verschillende adresseerbare formaten.

DATA CAPTURE
De manier waarop gegevens worden verzameld van externe bronnen, die op een of andere manier met de centrale computer zijn verbonden.

DEBUGGING
Het opsporen en corrigeren van fouten in een programma.

DECIMAAL SYSTEEM DECIMALE NOTATIE
Het getallensysteem met als basis 10, en cijfers van 0 tot 9 en weergegeven in tientallen, honderdtallen, duizendtallen, enz.

DIAGNOSTIEK
Een foutmelding die automatisch door de computer wordt gegeven als een fout wordt ontdekt.

DIGITAAL
Beschrijft de veranderende hoeveelheid in afgeronde stappen, anders dan als continu proces. Het tegenovergestelde van analoog.

DIGITISER
Een manier om analoge signalen in de computer in te voeren.

DISKETTE - FLOPPY DISK
Een uitneembare magneetschijf met een diameter van 3 of 5,25 inch, gebruikt voor opslag van gegevens. De dunne schijf is in een rechthoekig omhulsel gevat en kan duidelijk meer gegevens bevatten dan een cassette. Is bovendien veel sneller.

DISK DRIVE
Een apparaat dat gegevens leest en schrijft op een disk.

DOCUMENTATIE
De handleidingen, die met de hard- of software worden meegeleverd en verklaren hoe het gebruikt moet worden.

DOS
Disk Operating System. De software die alle functies van een diskdrive stuurt.

DOT MATRIX
Een rechthoekig puntenrooster waarmee men tekens kan uitbeelden.

EDITTEN
Het corrigeren of aanbrengen van wijzigingen in een bestand, gegevens of een programma.

EDITOR
Een programma normalerwijze in ROM, welke het mogelijk maakt het editten uit te voeren.

EPROM
Erasable Programmable Read Only Memory, een soort ROM dat door de gebruiker uitgewist en opnieuw geprogrammeerd kan worden. Het uitwissen gebeurt door de EPROM bloot te stellen aan ultra-violet licht van hoge intensiteit.

FILE
Een verzameling gegevens, meestal op cassette of disk opgeslagen.

FIRMWARE
Een programma dat in een ROM opgeslagen is. Een kruising tussen hard- en software.

FIXED POINT NUMBER
Een getal, waarvan de decimale punt in een vaste positie staat opgesteld, dus met een vast aantal cijfers achter de komma.

FLOATING POINT NUMBER
Een getal waarbij het aantal cijfers achter de komma kan varieren.

FORTH
Een zeer snelle programmeertaal, die qua snelheid en complexiteit ligt tussen een hogere programmeertaal en machinecode. Niet een taal voor beginners.

FUNCTIE TOETS
Een toets op een toetsenbord, die gebruikt wordt om een reeks instructies met een druk op de toets te geven. Meestal is de functie van deze toets vooraf vastgesteld.

GATE
Logische 'gates' laten het passeren van gegevens toe wanneer aan bepaalde voorwaarden voldaan is. Bijv. OR, AND, XOR enz.. Zie ook booleaanse algebra.

GRAPHICS
Weergave op het beeldscherm, anders dan tekst: bijv. lijnen, cirkels enz. Met een geschikte printer kan men dit ook printen.

GRAPHICS TEKEN (CHARACTER)

Bijzondere tekens of vormen, voor het maken van grafische figuren. De CPC464 heeft een gehele set die in aanhangsel III beschreven zijn.

GRAPHICS CURSOR

Gelijk aan de tekstcursor, maar alleen voor het grafisch beeldscherm. Bij de CPC464 onzichtbaar, maar nodig voor het tekenen van grafieken. Niet met de grafische tekens te verwisselen (zie aanhangsel III), die een deel van de tekenset vormen en die met de tekstcursor worden geprint.

GRAPHICS MODE

Vroegere microcomputers moesten in verschillende modi worden gezet voor verwerking van tekst of graphics. De moderne computers kunnen beide functies tegelijkertijd behandelen.

GRAPHICS TABLET

Een apparaat dat coordinaatpunten van een beeld of een tekening voor verwerking aan de computer geeft. Een soort analoog/digitaal omzetter.

HANDSHAKING

Een reeks van electronische signalen welke initiatie, controle en synchronisatie van de uitwisseling van gegevens tussen een computer en een randapparaat of tussen twee computers verzorgen.

HARD COPY

Papieruitdraai van een programma of andere tekst of een grafisch beeld. De tijdelijke weergave op het beeldscherm heet ´softcopy´.

HARDWARE

De electronische en mechanische componenten van een computersysteem. Alles dat geen firmware of software is

HEXADECIMALE NOTATIE

Getallen met als basis 16. Zie aanhangsel II. In de CPC464 wordt een hex getal aangegeven met & of &H.

HOGERE PROGRAMMEERTAAL

Programmeertaal die de normale spreekwijze tamelijk dicht benadert, zodat deze makkelijk geinterpreteerd kan worden. Langzamer dan machinetaalprogramma´s doch veel makkelijker te begrijpen. zoals BASIC.

IEEE-488

Een standaard aansluiting van apparaten op een computer. Ongeveer als de centronics parallel interface, maar niet geheel compatibel.

INFORMATIE TECHNIEK

Alles wat te maken heeft met electronica, informatieverwerking, communicatie: tekstverwerking, DATA communicatie, viditel enz.

INPUT - INVOER

Alles wat een computer wordt ingevoerd vanaf toetsenbord, cassette, disk, modem, serieele interface enz..

INSTRUCTIE
Een verzoek/opdracht die door de computer uitgevoerd moet worden. Een reeks van instructies vormen een programma.

INTEGER GETAL
Is een geheel getal, d.w.z. zonder breukdeel: +6, 0 en -234.

INTEGREERD CIRCUIT
Zie chip.

INTERACTIEF
Dialoog tussen gebruiker en het programma, waarbij de gebruiker gevraagd wordt om iets in te voeren. De actie van de gebruiker beinvloedt direkt de verdere afloop van het programma.

INTERFACE
De verbinding tussen de computer en de buitenwereld, zoals het toetsen-bord, beeldscherm, modem enz..

INTERPRETER
Een vertaalsysteem dat de instructies uit een hogere programmeer-taal bij de uitvoering van een programma omzet in machinecode.

I/O - INPUT/OUTPUT
Invoer/uitvoer.

ITERATIE
Een van de elementen van de computertechniek. De computer verdeelt alle opdrachten in kleine stukjes die kunnen worden verwerkt door de CPU. Daartoe moet de computer tussen vele kleine onderdelen heen en en weerspringen tot de opdracht is uitgevoerd.

JOYSTICK
Een invoer apparaat, dat over het algemeen de functie van de cursor toetsen vervangt en het spelen van spelletjes vergemakkelijkt.

K
Een afkorting voor ´kilo´:1000, kilobyte. In de computertechniek is is dat echter 1024 (decimaal), ontstaan uit het binaire stelsel nl. 2 tot de macht 10.

KEYWORD - BASICWOORD
Een woord dat in een computerprogramma een bepaalde bewerking uitvoert.

KLOK - CLOCK
Een tijdsysteem in de computer als basis voor het gelijklopen van interne functies. Een ´real time´ klok houdt de uren, datum enz. bij.

KUNSTMATIGE INTELLIGENTIE
Een structuur in programmeertechnieken waarin het programma ´leert´ van opgedane ervaringen.

LEAST SIGNIFICANT BIT
De meest rechtse bit in een binair getal (zie aanhangsel II).

LAGERE PROGRAMMEERTAAL
Assembler is een voorbeeld van een lagere programmeertaal. De instructies worden direkt omgezet in machinecode.

LICHT PEN
Een ander alternatieve invoer methode, maakt gebruik van een pen of 'toverstaf' !

LISP
Afkorting van LISt Processing, een hogere programmeertaal.

LOGICA
De electronische componenten, welke de elementaire logische operaties en functies uitvoeren, waaruit iedere computer opgebouwd is.

LOGO
Een eenvoudig te leren programmeertaal, die veel op scholen gebruikt wordt.

LSI
Large Scale Integration, de technologie waarbij grotere geintegreerde circuits op kleine stukjes silicium worden gemaakt.

LUS
Een serie instructies die herhaald uitgevoerd worden totdat, ofwel een bepaald aantal herhalingen bereikt is, ofwel het testresultaat is veranderd.

MACHINE CODE
De programmeertaal die door de computer rechtstreeks wordt begrepen omdat alle instructies uit binaire getallen bestaan.

MAN MACHINE INTERFACE
Het punt waarop interaktie tussen mens en computer plaatsvindt: toetsenbord, beeldscherm, geluid enz.

MATRIX
Een rangschikking van de punten die de tekens op het scherm vormen, of op de printkop van een 'dot matrix' printer. Ook: term in gebruik bij wiskunde en computertechniek die arrays omvat.

MEMORY — GEHEUGEN
Een gedeelte van de computer dat gegevens vast kan houden, die later weer gelezen kunnen worden. Meestal wordt deze term gebruikt voor het centrale geheugen van een computer.

MEMORY MAP — GEHEUGENPLAN
De opbouw van het centrale geheugen waarbij o.a. aangegeven wordt waar zich de besturingsroutines bevinden voor tape, disk, scherm, e.d.

MENU
Een lijst van verschillende mogelijkheden in een programmma die op het scherm getoond wordt, en waaruit een keuze kan worden gemaakt.

MICROPROCESSOR
Een chip die de machinecode bewerkingen uitvoerd. De microprocessor is het hart van een microcomputer.

MODEM
Een MOdulator-DEModulator die de computers-I/O verbindt met een telefoon of een ander serieele gegevens verzendingsapparaat. Zie acoustic coupler.

MONITOR
Het schermgedeelte van een computer, alsook een term waarmee een machinetaalprogramma bedoeld wordt.

MSB - MOST SIGNIFICANT BIT
Is het meest linkse bit in een binair getal.

MUIS - MOUSE
Een klein instrument dat door de hand over een glad oppervlak wordt bewogen en zo een cursor bestuurdt om over het scherm te bewegen.

NETWORK
Een systeem dat bestaat uit een of meerdere computers, waarbij communicatie via draad en/of modems plaatsvindt.

NIBBLE - NETWERK
Een eenheid van een halve byte, 4 bits.

NUMERIEK TOETSENBORD
Een verzameling numerieke tekens op sommige toetsenborden die in een rechthoekig blok gegroepeerd zijn, zodat numerieke gegevens efficienter ingevoerd kunnen worden.

OCR
Optical Character Recognition, het proces waarbij gegevens door een optische sensor gelezen en door de computer rechtstreeks verwerkt kunnen worden.

OCTAL
Getallenstelsel met basis 8, ieder cijfer (0-7) bestaat uit 3 bits.

OFF LINE
Een systeem of apparatuur, die niet rechtstreeks met de CPU verbonden is en daardoor niet onder besturing van de CPU valt.

ON LINE
Het tegenovergestelde van on-line.

OPERATING SYSTEEM
Software die het gebruik van het geheugen en de coordinatie van akties in een computer stuurt.

OUTPUT - UITVOER
Gegevens die afkomstig zijn van de computer, als resultaat van computer bewerkingen.

OPERATOR
Het gedeelte van een wiskundige uitdrukking dat gebruikt wordt om een getal de bewerking van een ander te laten ondergaan: +, -, *, /.

OVERSCHRIJVEN - OVERWRITE
Wissen van een geheugengebied door nieuwe gegevens in te voeren.

PADDLE
Een alternatief voor de joystick.

PARALLEL INTERFACE
De CPC464 heeft een aansluiting voor een parallel printer. Alle bits die een teken voorstellen, worden tegelijk over meerdere lijnen verzonden. Parallelle transmissie is sneller dan serieele, maar er zijn meer lijnen tussen de zendende en de ontvangende locatie nodig.

PASCAL
Een programmeertaal die genoemd is naar de Franse wiskundige Blaise Pascal (1623-1662). Wordt bijna altijd gecompileerd voordat er mee wordt gewerkt. Meestal de eerste taal na BASIC die iemand leert. Het voordeel is de veel betere programmeertechniek die men hiermee leert.

PEEK
Een BASIC instructie die het mogelijk maakt een waarde te lezen op een bepaalde plaats in het geheugen.

PIXEL - BEELDPUNT
Het kleinste gebied op een scherm dat door de hardware kan worden bestuurd.

PLOTTER
Een type printer dat lijnen trekt met pennen i.p.v. tekens opbouwd uit puntjes. Gebruikt voor het maken van tekeningen.

POKE
Een BASIC instructie die het mogelijk maakt een waarde te schrijven op een bepaalde plaats in het geheugen. Zie hoofdstuk 8.

POORT - PORT
Een speciaal adresseerbaar punt in de interface voor invoer of uitvoer van gegevens.

PORTABILITY
De mogelijkheid om dezelfde software van of op andere machines te gebruiken. Bijv. via CP/M.

PRINTER
Een ´hardcopy´ methode voor het printen van tekst.

PROGRAMMA - PROGRAM
De combinatie van instructies die de computer een opdracht laten uitvoeren. Dit kan alles zijn van een simpele routine tot complete applicatie-programma´s zoals tekstverwerking.

QWERTY TOETSENBORD
Een toetsenbord van een normale typemachine waarbij de letters q w e r t y links boven in het alfabetische gedeelte voorkomen.

RAM
Random Access Memory. Geheugen waar men, door gebruik te maken van de interne schakelingen van de computer, zowel uit kan lezen als naar kan schrijven.

RANDAPPARAAT - PERIPHERAL
Printer, modems, joysticks, diskdrive, kortom alles wat aan een computer kan worden aangesloten.

RANDOM NUMBER
Toevalsgetal. Een willekeurig getal dat samengesteld wordt door een willekeurige reeks cijfers.

RASTER
Het patroon dat gevormd wordt via het schrijven in een aantal horizontale lijnen op een beeldscherm.

REEEL GETAL
Een getal met cijfers aan beide zijden van de decimale punt. Een variabele kan toch een reeel getal bevatten, alhoewel het in een programma als een integer wordt uitgelezen.

REAL TIME
De tijd van het moment, de gebeurtenissen die worden weergegeven op het moment dat ze plaats vinden.

RECURSIEF
Een serie van herhaalde stappen binnen een programma, een routine waarbij het resultaat bij iedere herhaling in verbinding staat met het voorgaande. Een bekend voorbeeld: een routine die zichzelf aanroept.

REFRESH
Het actualiseren van gegevens op een beeldscherm of in een geheugen. De voorhanden zijnde gegevens hoeven daarbij niet te worden vernietigd.

REGELNUMMER
De volgordenummering van instructieregels in BASIC en enkele andere programmeertalen.

REGISTER
Een geheugeneenheid binnen de CPU voor tijdelijke opslag.

ROM
Read Only Memory. Alleen uit te lezen geheugen. Halfgeleider-geheugen dat eenmaal beschreven, niet gewist of overschreven kan worden.

ROUTINE
Een gedeelte van een programma dat een ´routine´ opdracht uitvoert. Een ´sub´ programma dat of in het hoofdprogramma zit of als aparte module bestaat voor invoeging in diverse applicatie programma´s. Bijv. een programma dat een klok voorsteld en steeds zijn informatie (iedere sec.) haalt bij de systeemklok, kan worden gezien als een routine.

RUIS - NOISE
De toongenerator van de CPC464 maakt het de gebruiker mogelijk met de SOUND instructie ruiseffekten te maken.

SCREEN EDITOR
Een programma voor teksten of programma's waarbij de cursor op iedere plaats van het beeldscherm kan worden gezet, om het daar staande teken te veranderen.

SCROLLING
Het omhoog schuiven van de tekst op het scherm om ruimte te maken voor een nieuwe regel.

SCHEIDER
Een scheidingsteken dat dezelfde functie heeft als een delimiter. De grens tussen gereserveerde woorden of andere elementen worden hiermee in een programma aangegeven.

SERIEELE INTERFACE
RS-232C: Een standaardsysteem voor communicatie naar en van computersystemen. Tekens worden bit voor bit overgezonden.

SIMULATIE
Een programma of apparaat dat op mathematische wijze een proces of systeem simuleert.

SPEECH RECOGNITION
De omvorming van gesproken woorden in instructies, die de computer herkennen kan.

SPRAAK SYNTHESIZER
Het opwekken van gesimuleerde spraak door hard- en software

SPREADSHEET
Een programma dat het toelaat rijen en kolommen van getallen te bewerken. Door het toekennen van functies kan een wijziging van een bepaalde plaats resulteren in een verandering in alle gerelateerde getallen.

SPRITE
Een teken, door speciale hard- of software gecreeerd, dat zich vrij over het beeld kan bewegen en toevallig verschijnt of verdwijnen kan

STACK
Een gedeelte van het geheugen dat toegewezen is voor het opslaan van informatie, maar waarvan alleen het laatst toegevoegde getal weer kan worden opgeroepen.

STATEMENT
Een instructie, of reeks van instructies in een computerprogramma.

STROOMSCHEMA - FLOWCHART
Schematische weergave van het programma en logische processen, waarmee de reeks van gebeurtenissen tijdens het RUNnen gevolgd kunnen worden.

TERMINAL
Een apparaat dat door een persoon gebruikt wordt om gegevens te verzenden en te ontvangen naar en van een computersysteem.

TOETSENBORD - KEYBOARD
Een samenstelling van alfanumerieke toetsen, die instructies en informatie kan doorgeven aan de computer.

TOONGENERATOR
Dat deel in een computer dat geluiden en ruis kan laten horen door middel van de soft- en hardware.

TRUNCATED - INKORTEN
Een getal of tekst wordt ingekort. Deze functie kan gebruikt worden bij afrondingen.

TRUTH TABLE - WAARHEIDSTABEL
Het resultaat van een logische operatie is of 'true' of 'false'. De computer geeft dit weer als 1 of 0. Een lijst met alle mogelijke uitkomsten van logische operaties (bijv. IF A>B THEN C) heet waarheidstabel.

UITDRUKKING - EXPRESSION
Een eenvoudige of gecompliceerde formule om in het programma berekeningen uit te voeren met gegevens

USER DEFINED KEY-UDK
De CPC464 heeft 32 toetsen die gedefinieerd kunnen worden om opgaven te voltooien. Iedere toets kan 32 tekens opnemen.

UNSIGNED NUMBER
Een getal zonder kenmerk, waaruit kan blijken of het getal negatief of positief is.

UTILITY
Een compleet gebruikers programma voor sortering, opsporen, controleren, copieren, enz. van gegevens

VARIABELE
Een geheugenplaats, voorzien van een naam, waarin waarden kunnen worden opgeslagen, die gedurende de uitvoering van een programma kunnen veranderen. Er bestaan numerieke en alfabetische variabelen.

INDEX

G = Grondbeginsel
H = Hoofdstuk
A = Aanhangsel